AD-A089 240   MCDONNELL AIRCRAFT CO ST LOUIS MO   F/G 1/3

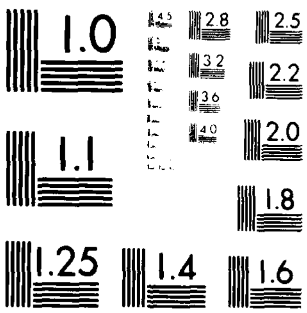
AIRCRAFT HYDRAULIC SYSTEMS DYNAMIC ANALYSIS. VOLUME VI. STEADY --ETC(U)

APR 80 R LEVEK, B YOUNG   F33615-74-C-2016

UNCLASSIFIED   AFAPL-TR-76-43-VOL-6   NL

1 OF 4

AD
A089240


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5

AFAPL-TR-76-43
VOLUME VI

LEVEL II ②

A038 6..

AD A089240

AIRCRAFT HYDRAULIC SYSTEMS DYNAMIC ANALYSIS

VOLUME VI: STEADY STATE FLOW ANALYSIS (SSFAN) COMPUTER PROGRAM TECHNICAL DESCRIPTION

McDonnell Aircraft Company
McDonnell Douglas Corporation
P.O. Box 516
St. Louis, Missouri 63166

April 1980

Technical Report AFAPL-TR-76-43, Volume VI
(This Report Supersedes AFAPL-TR-76-43, Volume VI, February 1977)
Final Report for Period June 1978 - November 1979

Aero Propulsion Laboratory
Air Force Wright Aeronautical Laboratories
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433

This technical report has been reviewed and is approved for publication.

Paul D. Lindquist
PAUL D. LINDQUIST
PROJECT ENGINEER
POWER SYSTEMS BRANCH
AEROSPACE POWER DIVISION

[illegible]
PAUL R. BERTHEAUD
TECHNICAL AREA MANAGER
POWER SYSTEMS BRANCH
AEROSPACE POWER DIVISION

FOR THE COMMANDER

Robert R Barthelemy
ROBERT R. BARTHELEMY
ACTING CHIEF, AEROSPACE POWER DIVISION
AERO PROPULSION LABORATORY

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFAPL-TR-76-43-VOL-~~VI~~ 6	2. GOVT ACCESSION NO. AD-A089240	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AIRCRAFT HYDRAULIC SYSTEMS DYNAMIC ANALYSIS VOLUME VI. STEADY STATE FLOW ANALYSIS (SSFAN) COMPUTER PROGRAM TECHNICAL DESCRIPTION		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report. June 1978-November 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ray Levek Bob Young		8. CONTRACT OR GRANT NUMBER(s) F3615-74-C-2016
9. PERFORMING ORGANIZATION NAME AND ADDRESS McDonnell Douglas Corporation P O Box 516 St. Louis, Missouri 63166		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 3145-30-18
11. CONTROLLING OFFICE NAME AND ADDRESS Aero Propulsion Laboratory (AFWAL/POOS) Air Force Wright Aeronautical Laboratories Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE April 1980
		13. NUMBER OF PAGES 348
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This report supersedes AFAPL-TR-76-43, Volume VI, February 1977.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Program	Flow Through Reservoir	Check Valve
Hydraulic System	Heat Exchanger	Flexible Hose
Steady State	Actuator	Accumulator
User Manual	Hydraulic Filter	Restrictor
Variable Delivery Pump	Solenoid Valve	Quasi-Transient

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SSFAN is a steady state hydraulic flow and pressure analysis computer program. Its primary purpose is to analyze non-linear resistance aircraft hydraulic systems. The program handles complex flow networks containing flow and/or pressure discontinuities such as unbalanced area actuators and check valves. Solutions for a combination of simultaneously operating subsystems are easily obtained. The program is designed using a building block approach so that new component or element models may be added with minimum change to

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

41[illegible]111

the main program. The solution method is a Matrix type, using iteration to obtain a final flow and pressure balance. The program internally corrects viscosities for pressure, determines whether flow is laminar, transition or turbulent for use of appropriate resistance factors and corrects reservoir pressure for altitude effects. A Quasi-transient section has been added to allow multiple steady state calculations when simulating subsystem operations. The data is stored and can be printed in either tabular form or computer plot form.

The program was written with the aircraft hydraulic system designer in mind. The terminology and units are commonly used terms such as fluid viscosity in centistokes, temperatures in degrees Fahrenheit and flow in gallons per minute. Conversion of units for calculation is accomplished internally in the program.

PREFACE

This Steady State Flow Analysis (SSFAN) Computer Program Technical Description AFAPL-TR-76-43 Volume VI of an eight volume sequence has been updated. The technical description was prepared by the McDonnell Aircraft Company, Design Engineering Power and Fluid System Department, McDonnell Douglas Corporation under contract F33615-78-C-2026, with supplemental agreement P0003.

The effort was sponsored by the Aero Propulsion Laboratory, Air Force Systems Command, Wright-Patterson AFB, Ohio under Project No. 3145-30-27 with AFWAL/POOS, and was under the direction of Paul Lindquist and William Kinzig.

The technical description covers work conducted from 1 June 1978 through 31 January 1980. At McDonnell, Neil Pierce directed the program and was Principal Investigator. Special acknowledgement is also given to J. B. Greene, R. J. Levek, R. E. Young, M. J. Stevens, B. G. Nolan, and P. C. McAvoy.

PRECEDING PAGE BLANK-NOT FILMED

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT)

TABLE OF CONTENTS (CONT)

SECTION TITLE PAGE

TABLE OF CONTENTS (CONT)

TABLE OF CONTENTS (CONT)

TABLE OF CONTENTS (CONT)

LIST OF FIGURES

LIST OF FIGURES

SECTION I

INTRODUCTION

The SSFAN computer program is currently mechanized for the CDC 6500/6600 computer. It is coded in Fortran IV and is used to simulate aircraft hydraulic fluid flow systems, which typically consist of multiple branches. The flow is assumed to be one dimensional and steady state. The effects of fluid viscosity and density changes with temperature and pressure are included in calculations. Altitude effects are also included. The output of SSFAN is predicted values of flow rate, pressure, subsystem operation time, etc.

Some of the element models comprising the program were derived using empirical data to describe the flow pressure-drop relationship. However, the data available is generally for one test temperature (room temperature), so that an extrapolation to extremely high or low temperatures may decrease the accuracy of the results. Other element models were derived from the theoretical mathematical equations. In some cases, the models contain correction factors based on previous test data.

SSFAN has been updated to extend its calculation capability and reduced in size by simplifying the overall organization of the program. The addition of the Quasi-transient model allows prediction of flow, pressure and actuator stroke versus time. The capability to input various fluid temperatures at components giving a temperature distribution throughout the system has been added.

A computer graph subroutine has been added to plot the quasi-transient data. Data may also be printed out in tabular form.

SECTION II

TECHNICAL SUMMARY

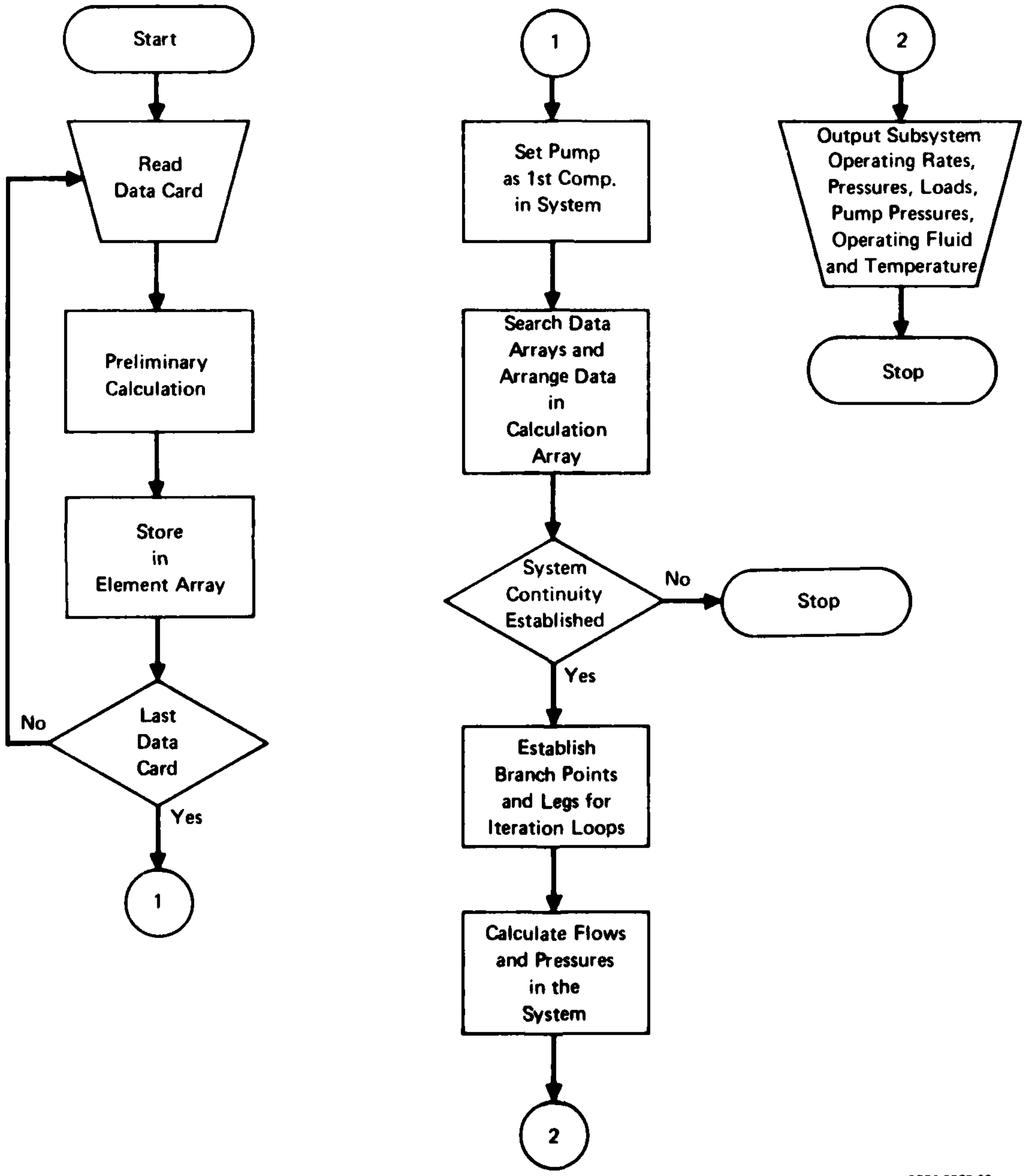
Input data cards containing the detailed element (component) data and system location identifiers (junction numbers) for each element in the system are established. The data is read into 2 storage arrays. Some data, such as tubing bends, are calculated for energy loss coefficients as the data is read in, and stored in the array with tube data. Fluid properties and aircraft altitude are also input. The altitude is used for reservoir pressure correction. The general SSFAN flow chart is shown in Figure 1 .

The pump or pumps are the first components in the system, unless an accumulator is designated as the pressure source. A search routine then searches the data arrays and builds the system by legs until the total system is built. If the input data is erroneous so that system continuity cannot be established, error messages are output and the run stops.

As the system is being built, resistance coefficients are calculated for each element and summed for a hydraulic system leg. They are stored in a leg array for the duration of the run.

Viscosity at atmospheric pressure is input to the program and is corrected initially for pressure. Resistance coefficient are established for all branch legs in the system from which the branch leg conductances are calculated. These resistance coefficients are established, based on whether the flow in the branch leg is in the laminar, transition or turbulent flow regime.

Branch points are established from an array which contains the active branch legs in the system. The system is first balanced using an initial guessed flow.



Start
Read Data Card
Preliminary Calculation
Store in Element Array
Last Data Card
No
Yes
1
1
Set Pump as 1st Comp. in System
Search Data Arrays and Arrange Data in Calculation Array
System Continuity Established
No
Stop
Yes
Establish Branch Points and Legs for Iteration Loops
Calculate Flows and Pressures in the System
2
2
Output Subsystem Operating Rates, Pressures, Loads, Pump Pressures, Operating Fluid and Temperature
Stop

GP74-0772-29

FIGURE 1
SSFAN FLOW CHART

The general solution technique is based on an iterative method using a matrix to solve for pressures at branch points in the system. Each branch leg resistance is described by a nonlinear equation as a function of flow in the leg. This equation is solved for the leg pressure drop which is then used to obtain a leg conductance constant at the current flow rate. Finally conductances are placed in the matrix for a new solution and the iteration is continued until two successive solutions of the matrix provide flow balances for all branch legs within .001 gpm.

The SSFAN program may be run in a quasi-transient mode. With an input of time interval, time step (optional) and some additional element data, the quasi-transient model may be used to predict pressures, flows, subsystem operating times, etc. If a time step is not input a default value of 100 steps is used which will provide a full set of data points for the graph subroutine.

SECTION III

SSFAN MAIN PROGRAM

The Main Program is responsible for the overall program setup and execution. It controls the entire operation of the SSFAN program from data initialization to output. Any error conditions recognized by the program terminates that section of the program after printing out error messages.

The DATE subroutine is called, and the data is read-in via the input section of the main program. When the input is complete, a viscosity - pressure correction factor is calculated for the hydraulic fluid, and viscosity and ambient pressure are computed for the system temperature and altitude.

After these parameters are found, system assembly begins. During the course of leg building static resistance coefficients are calculated for each leg in the system. A check is made to determine whether the program is being run in the quasi-transient mode. If not, the system is solved for pressures and flows and results are printed according to the selected output. If the program is run in the quasi-transient mode, the initial flow balance is printed out for all junctions, but only the quasi-transient user selected output is printed for the remainder of the run.

See Figure 2 for the SSFAN Main Program Flow Diagram.

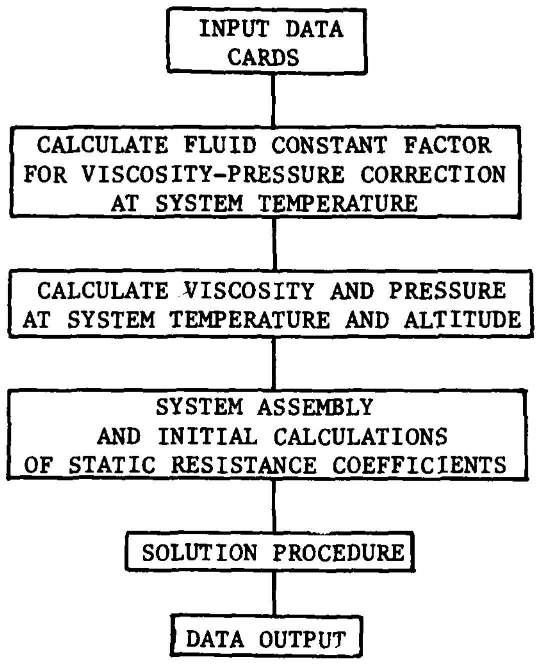

INPUT DATA CARDS
CALCULATE FLUID CONSTANT FACTOR FOR VISCOSITY-PRESSURE CORRECTION AT SYSTEM TEMPERATURE
CALCULATE VISCOSITY AND PRESSURE AT SYSTEM TEMPERATURE AND ALTITUDE
SYSTEM ASSEMBLY AND INITIAL CALCULATIONS OF STATIC RESISTANCE COEFFICIENTS
SOLUTION PROCEDURE
DATA OUTPUT

Figure 2
SSFAN Main Program
Flow Diagram

3.1 Description of SSFAN Operation

A call is made to the computer for the Julian date, which is stored in BLK8. The first major section of the SSFAN program is the data input section which reads in the data cards. If an error occurs on input, IERROR will be reset to a value other than zero and the program is terminated. A call is then made to VISD which returns the viscosity and density values in BLK1. The title page of the SSFAN program is output with the call to the subroutine OPUT4. If, because of erroneous data points a negative viscosity should be calculated, the program will stop and print out an error message. The user may select to have his data deck printed out exactly as read in to check for errors (see Output Options). The ambient atmosphere pressure for the input altitude is calculated and put in BLK1 under the name of PAMB.

The second major part of the program begins with the call of the SDSORT subroutine to set up arrays for leg building. BUILD subroutine assembles the legs. If all the legs are assembled with no errors generated, pressure points are assigned to the ends of the legs. The SDSORT and BUILD subroutines, called from the Main Program, control the entire assembly phase in the SSFAN program.

The calculation of the system pressures and flows in the next major part of SSFAN is totally controlled by the CALC subroutine. Once the solution phase is finished the information is output through the user selected output.

Three arrays that are critical to the solution procedure of SSFAN are BLEG, ILEP, and PQL. All of these arrays are generated by the assembly phase.

BLEG and PQL are updated in the computation section and contain the final solutions to be used by the Output subroutines. A knowledge of the information contained in these arrays is essential to an understanding of the SSFAN program operation. SSFAN Sample Case Number 1, a two actuator subsystem, is used to explain how the elements are entered and arranged in BLEG, ILEP and PQL, Figure 3 .

The ports of every element have a junction number assigned to them. The assembly phase of SSFAN matches junction numbers of the system elements until a leg in the system is assembled. The leg is identified with a leg number and assigned pressure point numbers to the upstream and downstream ends. The legs are built from pressure point to pressure point. These are the points at which pressures are calculated and printed out for the system. Elements that contain pressure points in the SSFAN program are pumps, accumulators, reservoirs, actuators, tees, crosses, valves and motors. Figure 4 is a schematic diagram of how leg numbers and pressure points are assigned by the assembly phase of SSFAN to the SSFAN Sample Case Number 1.

From the assignment of leg numbers and pressure points, the ILEP array is built. The pressure points at the up and downstream locations of the leg are written into columns one and two, respectively. An ILEP array for the system in Figure 3 is shown in Figure 5 .

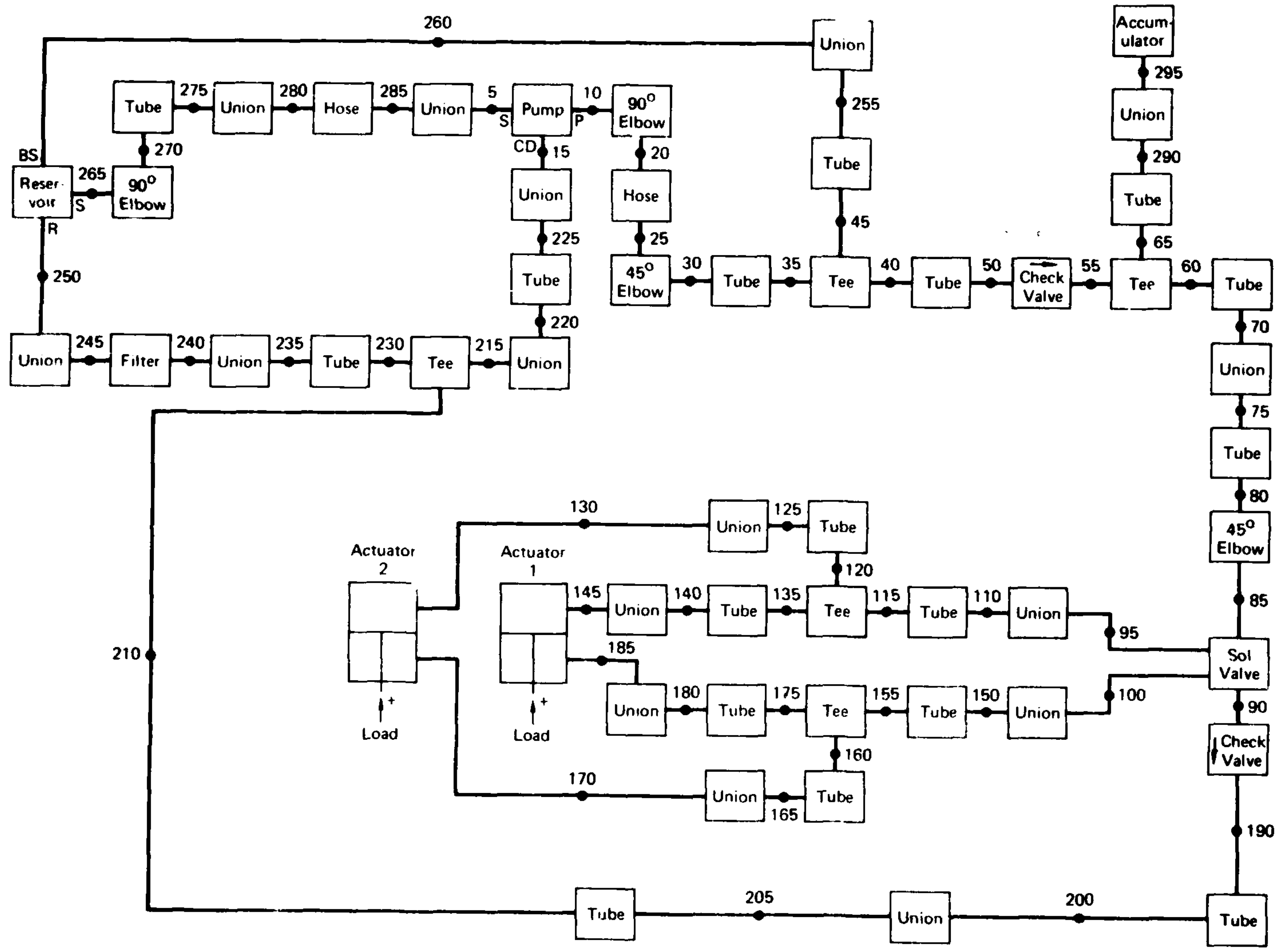
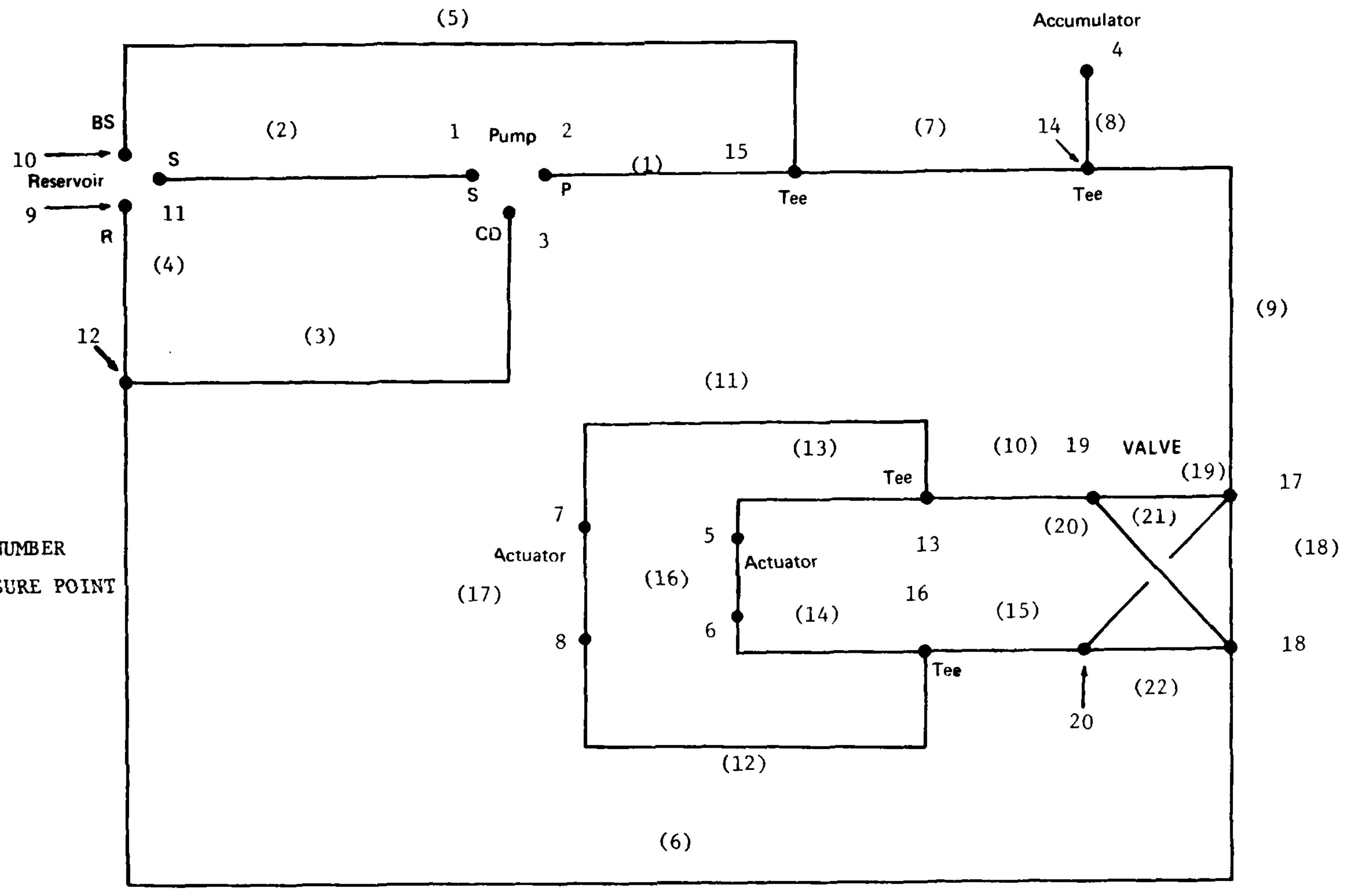


FIGURE 3
SSFAN SAMPLE CASE NUMBER 1
2 Actuator Subsystem



(X) LEG NUMBER

X PRESSURE POINT

FIGURE 4

System Schematic Diagram SSFAN Sample Case No. 1

Leg Numbers	Upstream Pressure Point	Downstream Pressure Point
	ILEP COL 1	ILEP COL 2
1	2	15
2	1	11
3	3	12
4	9	12
5	10	15
6	12	18
7	15	14
8	14	4
9	14	17
10	19	13
11	13	7
12	8	16
13	13	5
14	6	16
15	16	20
16	5	6
17	7	8
18	17	18
19	17	19
20	19	18
21	17	20
22	20	18

FIGURE 5
ILEP Array
For SSFAN Sample Case Number 1

Thus the ILEP array contains the description of how the system is assembled.

The PQL array has as many rows as there are pressure points in the system. For example, Case Number 1, Figure 4 shows a total of 20 pressure points. Column one of the PQL has the values of pressure at these points. Pressure points in a system are termed as being either constant pressure points or branch points. A constant pressure point must be at an end point in a system, but all end points need not be constant pressure points. The values of pressure at constant pressure points are treated as constants for a solution during an iteration, but are updated for the next iteration.

A branch point thus is a pressure point that is calculated and it may also be an end point or a point at which two or more flows meet.

Column two of the PQL array contains the flow loss or gain at a branch point or end point due to a dynamic element. The element type number for the pressure point is in column three. Columns 4, 5, 6 and 7 contain the junction numbers of the pressure point element. Column 8 contains the fluid temperature at the pressure point. Figure 6 gives the initial PQL array for the SSFAN Sample Case Number 1. Initial constant pressures are set at three thousand PSI for the pump pressure port and fifty psi for the reservoir return and suction ports. The branch points are all minus ones. The values for all the pressure points are updated as the solution procedure iterates to the final answers. All the terms in column two are initially set to zero but these too are updated in the iteration process.

Column Number / Pressure Point	Press 1	QGain or QLoss 2	Type 3	JCT1 4	JCT2 5	JCT3 6	JCT4 7	Fluid Temp 8
1	-1.	0.	5.	5.	0.	0.	0.	100.
2	3000.	0.	5.	10.	0.	0.	0.	100.
3	-1.	0.	5.	15.	0.	0.	0.	100.
4	-1.	0.	7.	295.	0.	0.	0.	100.
5	-1.	0.	4.	145.	0.	0.	0.	100.
6	-1.	0.	4.	135.	0.	0.	0.	100.
7	-1.	0.	4.	130.	0.	0.	0.	100.
8	-1.	0.	4.	170.	0.	0.	0.	100.
9	50.	0.	9.	250.	0.	0.	0.	100.
10	-1.	0.	9.	260.	0.	0.	0.	100.
11	50.	0.	9.	265.	0.	0.	0.	100.
12	-1.	0.	24.	210.	215.	230.	0.	100.
13	-1.	0.	24.	115.	120.	135.	0.	100.
14	-1.	0.	24.	55.	65.	60.	0.	100.
15	-1.	0.	24.	35.	45.	40.	0.	100.
16	-1.	0.	24.	175.	160.	155.	0.	100.
17	-1.	0.	34.	85.	0.	0.	0.	100.
18	-1.	0.	34.	90.	0.	0.	0.	100.
19	-1.	0.	34.	95.	0	0.	0.	100.
20	-1.	0.	34.	100.	0.	0.	0.	100.

FIGURE 6

PQL Array for SSFAN Sample Case No. 1

The BLEG array contains the parameters for the solution procedure. Each row in BLEG represents a leg in the system. The columns contain the information particular to each leg. Some of the data in the columns of BLEG for the Sample Case Number 1 are presented in Figure 7.

Leg Number	Column Number								
	1	2	3	4	5	6	7	8	9
1	10.00	35.00	12.8514	.6716	-1.9080	1.2468	12.8514	.0001	.2122
2	5.00	265.00	-13.5113	.8979	0.0000	5.3890	-13.5113	.0001	.0575
3	15.00	215.00	.6605	.2088	2.2863	.0884	.6604	.0001	12.3551
4	250.00	230.00	-9.8683	.4000	0.0000	.1271	-9.8694	.0001	.2230
	Upstream Junction Number Ref: Sec 5.1.2	Downstream Junction Number Ref: Sec 5.1.2	Updated Flow Rate in Leg Variable Name: Q Units: GPM Ref: Sec 6.1	Average Diameter of the Leg Includes Correction for Orifices in the Leg Ref: Sec 5.1.5	Pressure Gains or Drops in a Leg Due to a Dynamic Subroutine Variable Name: PD Units: PSI Ref: Sec 6.1 and 6.2	Leg Conductances from Equation $G = \frac{Q}{\Delta P}$ Variable Name: G Ref: Sec 6.1 Note: In Subroutine TTL (Sec 6.1.3) the ΔP Values are Written into this Position, the CALC Subroutine (Ref 6.1.1 and 6.1.2) Convert this Column to a G Value	Latest Calculated Flow Rate Variable Name: QNEW Units: GPM Ref: Sec 6.1 Note: On Assembly this Column Contains any Orifice Diameters in the Leg Ref: Sec 5.1.5	Constant Term of Pressure Drop Equation Generated for the Leg Ref: App B Note: Value Assigned in Block Data	Constant Term of the Q for Pressure Drop Equation of the Leg Ref: App B, Sec 5.2

GP75 0270 47

FIGURE 7
PARTIAL BLEG ARRAY FOR
SSFAN SAMPLE CASE NUMBER 1

Column Number						
10	11	12	13	14	15	16
.0802	.0165	0.0000	151.54	100.0	.00008	14.60000
.0175	.0046	0.0000	127.70	100.0	.00008	14.60000
11.2655	1.5543	0.0000	80.78	100.0	.00008	14.60000
.0979	.0221	7.1027	73.36	100.0	.00008	14.60000
Constant Term of $Q^{1.75}$ for Pressure Drop Equation of the Leg Ref: App B Sec 5.2	Constant Term of Q^2 for Pressure Drop Equation of the Leg Ref: App B and Sec. 5.2	Variable Pressure Drop Term Due to Elements in Pressure Drop Equation of the Leg Ref: App B and Sec 5.2	Actual Leg Length (in.)	Average Fluid Temperature in Leg	Fluid K for Fluid in Leg	Leg Average Fluid Viscosity (cs)

FIGURE 7 (Continued)

GP75 0270 19

3.2 Math Model

The approach to the SSFAN method of analysis is based on the principle of conservation of energy which results in an energy balance of:

$$\text{Energy at Section 1} + \text{Energy Added} - \text{Energy Lost} - \text{Energy Extracted} = \text{Energy at Section 2}$$

In the direction of flow from 1 to 2 for steady state flow of incompressible fluids in which the change in internal energy is negligible, this may be written in a form of the Bernoulli theorem or equation as:

$$\frac{P_1}{\rho_1} + \frac{V_1^2}{2g} + Z_1 + H_A - H_L - H_E = \frac{P_2}{\rho_2} + \frac{V_2^2}{2g} + Z_2 \quad (1)$$

Further simplifications may be made if the density is assumed to be the same at points 1 and 2.

therefore $\rho_1 = \rho_2 = \rho$

In an aircraft hydraulic system where the lines are relatively small, the pump(s) centrally located and for steady state flow operation under high pressure, the difference in elevation between points 1 and 2 ($Z_1 - Z_2$) may be ignored. However the reference altitude for the low pressure pump suction system can not be ignored as noted in the reservoir mathematical analysis.

The energy added (H_A) is considered to be in the form of a pressure rise in the system at a pump. Energy lost (H_L) is the frictional and viscous pressure drops due to flow and energy extracted (H_E) is at an actuated subsystem moving against a resistance such as a flight control surface actuator. This may also be equated to equivalent pressure loss in the system. It should be noted that in the case of a flight control surface, an aiding load on the actuator actually adds energy to the system and is accounted for in this analysis. The method of analysis outlined below is similar to Reference (1).

Rewriting the Bernoulli equation considering the above conditions,

$$\frac{P_1}{\rho} + \frac{V_1^2}{2g} - H_{12} = \frac{P_2}{\rho} + \frac{V_2^2}{2g} \qquad (2)$$

or

$$P_1 + \frac{\rho V_1^2}{2g} - P_{12} = P_2 + \frac{\rho V_2^2}{2g} \qquad (3)$$

Where: P = pressure PSI)

ρ = weight density (LB/IN^3)

V = fluid velocity (IN/SEC)

g = gravitational acceleration (IN/SEC/SEC)

$P_{12} = \rho H_{12}$ = friction loss, pump pressure rise, gain or loss of actuators (PSI)

A common term for hydraulic flow is

Q in gallons per minute (GPM)

To establish a simple relationship between the resistance from 1 to 2 and flow the resistance is defined as R_{12}, where the total loss in pressure from 1 to 2 is

$$R_{12}Q \qquad (4)$$

where R_{12} = resistance (PSI-MIN/GAL)

Then equations (3) and (4) may be combined:

$$R_{12} = \frac{1}{Q} \frac{\rho(V_2^2 - V_1^2)}{2g} + \Delta P_{12} \qquad (5)$$

The velocity term may also be written as

$$V = \frac{231}{60} \times \frac{Q}{A}$$

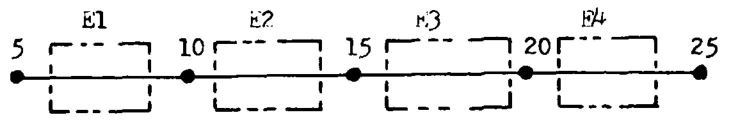
with V (IN/SEC)
Q (GPM)
A (IN)

Since $Q_1 = Q_2$ then the first term in the brackets of equation (5) drops out and leaves

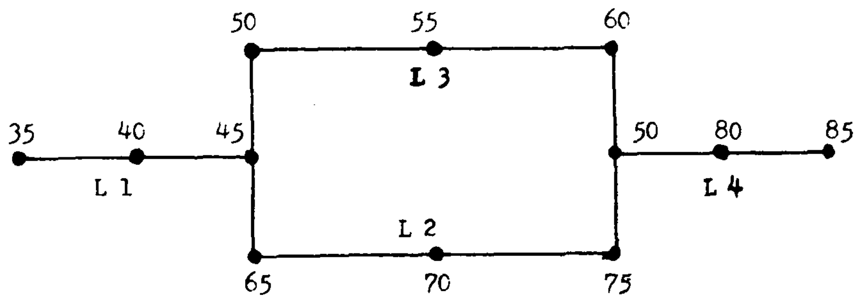
$$R_{12} = \frac{P_{12}}{Q}$$

This term then gives a linear relationship between P_{12} and Q required for the Matrix solution.

To correlate this to the system being analyzed, the following definitions are used. A system is described as shown in Figure 8.



BRANCH LEG WITH 4 ELEMENTS AND 5 JUNCTION POINTS



SYSTEM WITH 4 BRANCH LEGS AND 2 BRANCH POINTS

FIGURE 8

Junction Points, Elements, Branch Legs and Branch Points

The actual branch point in the system may best be seen by the illustration of the tee fitting, which is used as a branch point element. (see Figure 9)

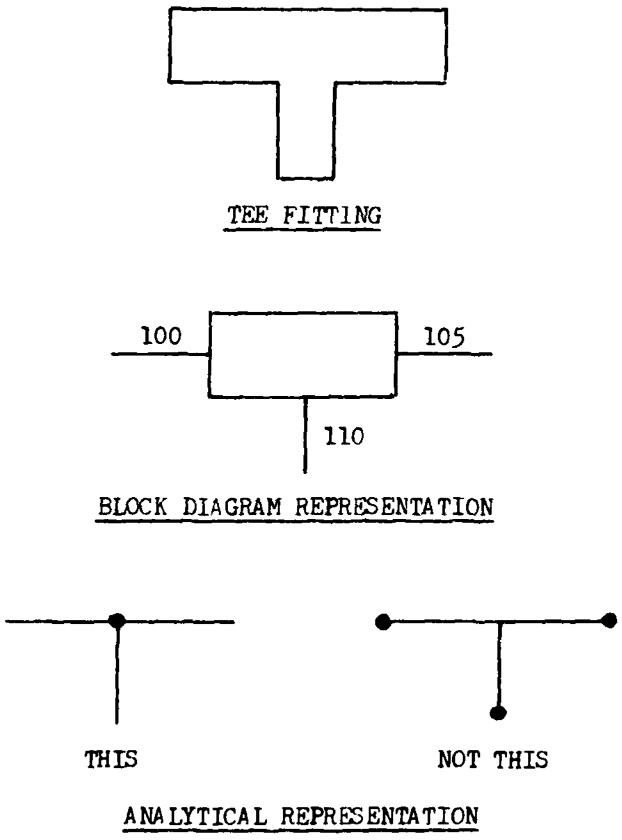


FIGURE 9

Tee Fitting Illustration

Figure 2 shows a branch leg with 4 elements (E1,E2,E3,E4) and 5 junction points (5,10,15,20,25). Also in Figure 7, a system with 4 branch legs (L1,L2,L3,L4) and 2 branch points (45,50). If a junction point is connected to only one other junction point, it is an end point.

The individual element resistances are summed along the branch legs to give a total resistance. A portion of a branch point type element resistance is allotted to each of its connecting branch legs. If a junction point is connected to only one other junction point, it is an end point.

The fluid conductance in a system is defined as the inverse of the resistance or

$$G_{12} = \frac{1}{R_{12}} \qquad (6)$$

where G_{12} = conductance between points 1 and 2

Equation (1) may be rewritten as

$$Q = G_{12} \, \Delta P_{12} \qquad (7)$$

Equation (7) is now in the form required for the Matrix solution subroutine SIMULT of Section 4.8. Equation (7) states that the flow, Q, in a branch leg of a hydraulic system equals the conductance (G_{12}) of the leg times the pressure difference between the upstream and downstream branch points (1 and 2).

In the current SSFAN hydraulic system analysis, branch points are considered to be located in tees, crosses, actuators and valves. End points are established at pumps, accumulators, reservoirs and motors.

The net flow at any branch point must be equal to zero to satisfy the continuity equation.

This requirement is satisfied at the Ith point if:

$$J \quad G_{IJ}(P_I - P_J \pm P_{IJ}) - \sum_K \pm Q_{IK} = 0 \qquad (8)$$

where

P_I = pressure at branch pointI

P_J = pressure at branch point J

P_{IJ} = A pressure rise or loss (from a pump or actuator) in branch leg IJ

Q_{IK} = Fixed flow in branch leg I_K connected to branch point I

When looking at any branch point I, the flows are summed for all branch legs connected to I not having fixed flow rates (legs IJ) plus branch legs connected to I having fixed flow rates (legs IK)

The signs of P_{IJ} and Q_{IK} are positive or negative depending on the direction of pressure loss or gain and flow loss or gain. The convention established is that the sign is positive if there is a pressure gain in a leg or a flow gain at a branch point.

The set of equations derived from equations (7) and (8) for a complex hydraulic flow network are set up in the SIMULT solution Matrix as follows.

$$\begin{bmatrix} g_{11} & g_{12} & \cdot & \cdot & \cdot & g_{IN} \\ g_{21} & g_{22} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{NI} & \cdot & \cdot & \cdot & \cdot & g_{NN} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \cdot \\ \cdot \\ \cdot \\ P_N \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \cdot \\ \cdot \\ \cdot \\ C_N \end{bmatrix}$$

where N = total number of branch points

M = total number of end points

$$g_{II} = \sum_{J=1}^{N+M} G_{IJ}$$

$$g_{IJ} = -G_{IJ}$$

$$C_I = \sum_{\substack{J=1 \\ J\ K}}^{N} \pm G_{IJ} \, \Delta P_{IJ} + {}_K \pm Q_{IK} \tag{9}$$

$$+ \sum_{\substack{L=N+1 \\ L\ K}}^{N+M} G_{IL} \qquad P_L \pm \quad P_{IJ}$$

P_I = pressure at a branch point

P_L = fixed pressure at an end point

Gaussian elimination with pivoting is used to solve the Matrix system of equations for P_1 to P_N.

The solution of the set of equations is accomplished in 4 basic steps:

(1) An initial guess is made for values of the flow rates in all the branch legs with non-fixed flow rates of the system. Experience has shown that this initial guess is not critical for the SSFAN solution and this guess is made internally in the program.

(2) In each branch leg with a non-fixed flow rate the current estimate of the flow rate in that leg is used to compute the conductance of the branch leg. The leg pressure drop (resistance) is calculated from a general equation of the form

$$DP(R) = K_1 + K_2Q + K_3Q^{1.75} + K_4Q^2 \qquad (10)$$

where the resistance coeffieients were defined when the branch legs were assembled, and Q is the current flow estimate.

DP = pressure drop (PSI)

Q = flow (GPM)

K_1, K_2, K_3, K_4 = resistance coefficient

DP's are calculated for all branch legs in the system from which a current value of G is calculated

$$G = \frac{1}{R} \text{ or } \frac{1}{DP} \qquad (11)$$

(3) These values of G are placed in the Matrix and the Matrix is solved for pressures at all branch points.

(4) The branch point pressures are then used to solve for a new flow rate where

$$Q_{NEW} = (P_I - P_J \pm \Delta P_{IJ})\, G_{IJ} \qquad (12)$$

Using this calculated value of Q_{NEW} along with Q_{OLD}, a new value of Q_{IJ} is calculated by the following equation

$$Q_{IJ} = \frac{Q_{OLD} + Q_{NEW}}{2} \qquad (13)$$

Q_{OLD} = previous flow value

Q_{NEW} = latest flow value

Q_{IJ} = the new guess flow rate-after an iteration Q_{IJ} becomes Q_{OLD}

Steps 2 through 4 are repeated until successive estimates in all branch legs agree within the tolerance specified.

From experience to date, one finds that equation (13) is the most optimal for updating the flow guess.

<u>Convergence Criteria</u>

The solution for flows in all the branch legs are final when all the previous flows and the latest calculated flows are within a specified tolerance. SSFAN tests the convergence of flows for two conditions; (1) Flows less than one gpm and (2) flows greater than 1 gpm. If both the old and new flows are less than one gpm equation (14) must be satisfied.

$$Q_{OLD} - Q_{NEW} \leq .001 \qquad (14)$$

For flow greater than or equal to one gpm equation (15)

$$\frac{Q_{OLD} - Q_{NEW}}{Q_{BIG}} \leq .001 \qquad (15)$$

Q_{BIG} = LARGER OF Q_{OLD} or Q_{NEW}

If equations (14) or (15) are not satisfied in each branch leg of the system for a specified number of iterations, the solution process will stop and an error message will be printed out indicating the failure to converge due to excessive iterations.

The calculations occurring in SSFAN Main Program pertain to the FLUIDK factor for a viscosity-pressure correction and a PAMB term for the pressure at the system altitude. The derivation of FLUIDK is found in Appendix B of this manual. The description of a pressure-altitude equation results from an application of altitude-temperature and pressure-temperature relationships for air, which is assumed to be a perfect gas. Reference (6) is the source for this equation which for completeness is derived below.

The change in temperature with altitude is constant over certain ranges. If the altitude above ground level is called h, then a constant termed the lapse rate λ is defined by the following equation:

$$\frac{dT}{dh} = \pm \lambda \qquad (16)$$

Where: T = Temperature

h = Height

λ = Lapse Rate

The positive sign in Equation (16) is used when the temperature increases with increasing altitude and the negative sign is used when the temperature decreases for increasing altitude.

The atmospheric pressure decreases with increasing altitude and may be expressed as follows:

$$\frac{dp}{dh} = -\rho g \qquad (17)$$

Where: p = Pressure (lb/ft^2)

h = Altitude (ft)

ρ = Density (slug/ft^3)

g = Gravitation Constant (ft/sec^2)

Assume that air obeys the perfect gas equation of state (18);

$$\frac{p}{\rho} = RT \qquad (18)$$

Where: p = Pressure (lb/ft^2)

T = Temperature (°R)

R = Gas Constant (ft lb$_f$/(slug)(°R))

ρ = Density (slug/ft^3)

Then substituting Equation (18) into Equation (17) one obtains:

$$\frac{dp}{p} = -\frac{dh}{RT}\, g \qquad (19)$$

Further combining Equation (16) with (19) gives:

$$\frac{dp}{p} = \mp \frac{g}{\lambda R}\, \frac{dT}{T} \qquad (20)$$

Integrating both sides of Equation (20) between the limits P_o to P and T_o to T one gets a solution of the form:

$$\left(\frac{P}{P_o}\right)^{\mp \lambda R/g} = \frac{T}{T_o} \qquad (21)$$

The relation between pressure and temperature for a perfect gas can be written as:

$$\left(\frac{P}{P_o}\right)^{(n-1)/n} = \frac{T}{T_o} \qquad (22)$$

Comparing exponents in Equations (21) and (22) one is able to write an expression for the lapse rate λ as being:

$$\lambda = \mp \frac{(n-1)g}{nR} \qquad (23)$$

Solving equation (22) for $\frac{n-1}{n}$ below:

$$\frac{n-1}{n} = \frac{\log \frac{T}{To}}{\log \frac{P}{Po}} \qquad (24)$$

From the ICAD standard atmosphere at sea level

$To = 518.688 \quad °F$

$P_O = 2116.22 \quad LB/FT^2$

at 10,000 FT

$T = 483.026 \quad °F$

$P = 1455.33 \quad LB/FT^2$

Substituting these values into equation (24) one finds that n = 1.23496. Letting g = 32.2 FT/SEC^2 and R = 1,716 FT^2/SEC^2 °R, λ from equation (23) has a value of approximately .00357°R/FT.

Equation (16) may be integrated to obtain a relation between altitude and temperature;

$$T - T_o = \pm \lambda (h - h_o) \qquad (25)$$

From Equations (21) and (25) pressure and altitude are both expressed in terms of temperature. An equation may now be written by combining (21) and (25) to solve for the pressure at altitude in terms of temperature and altitude.

Solve Equation (21) in terms of T and substitute into Equation (25). Dividing by $\pm \lambda$ one obtains:

$$h - h_o = \pm \frac{T_o}{\lambda} \left[\left(\frac{P}{P_o} \right)^{\lambda R/g} -1 \right] \qquad (26)$$

To simplify Equation (26) let $h_o = 0$ altitude, and then rearranging, the equation now reads:

$$\left(\frac{P}{P_o} \right)^{\lambda R/g} = \pm \frac{h\lambda}{T_o} + 1 \qquad (27)$$

or, rewriting (26), using the minus sign for the lapse rate to mean decreasing temperature with increasing altitude one may write:

$$P^{\lambda R/g} = P_o^{\lambda R/G} \left(1 - \frac{h\lambda}{T_o} \right) \qquad (28)$$

Further simplification of Equation (28) yields:

$$P = \left[P_o^{\lambda R/g} \left(1 - \frac{h\lambda}{T_o} \right) \right]^{g/\lambda R} \qquad (29)$$

At sea level with $h_o = 0$

$P_o = 14.7 \ lb_f/in^2$

$T_o = 519°R$

$\lambda = .00357 \ °R/ft$

$R = 1716 \ ft^2/(sec^2)\ (°R)$

$g = 32.2 \ ft/sec^2$

Equation (29) now becomes:

$$P = [1.6676(1 - h/145378.)]^{5.256} \qquad (30)$$

Equation (30) is used in the Main Program SFAN to calculate the pressure at any altitude between 0 and 36,089 ft. The P value is stored in the labeled common named BLK1 under the variable name PAMB. This makes the pressure value available to the component subroutines that are altitude dependent such as the reservoir subroutine.

Because of the abrupt slope changes of the temperature-altitude relationship, it is not possible to express the atmospheric characteristics as a continuous function. Different equations apply in each thermal layer. For altitudes of 36,089 to 65,000 ft. the atmospheric temperature is constant. To obtain a pressure altitude equation at these altitudes integrate equation (19) as follows:

$$\int_{P_1}^{P} \frac{dP}{P} = - \int_{h_1}^{h} \frac{dh}{RT} g \qquad (31)$$

or

$$LN\left(\frac{P}{P_1}\right) = g\,\frac{h_1 - h}{RT} \qquad (32)$$

where	P	=	pressure at altitude PSI
	P_1	=	pressure at base altitude PSI
	h	=	altitude (FT)
	h_1	=	altitude at base layer (FT)

Solving equation (31) for P:

$$P = p_1 e \left[\frac{g(h_1-h)}{RT}\right] \qquad (33)$$

using the ICAD standard atmosphere

$$P_1 = 472.679 \text{ LB/FT}^2 * \frac{1 \text{ FT}^2}{144 \text{ IN}^2} = 3.2835 \text{ PSI}$$

$$h_1 = 36{,}089 \text{ FT}$$

$$R = 1716 \text{ FT}^2/\text{SEC}^2 \ {}^\circ\text{R}$$

$$T = 389.988 \ {}^\circ\text{R}$$

$$g = 32.2 \text{ FT/SEC}^2$$

$$p = 3.2825 * \text{EXP}(4.806 \times 10^{-5} (36089-h)) \qquad (34)$$

Equation (34) is used in SSFAN for the pressure-altitude equation for 36,089 to 65,000 FT.

To define a relation for the 65,000 to 150,000 FT region of the atmosphere, the orthogonal polynominal method was used to fit least-squares polynomials to data. A third degree equation (35) was derived from the 1962 standard atmosphere data to obtain a reasonable correlation to the model atmosphere.

$$\text{PAMB} = [(867.42375-1.990498\text{E-}02(h)+1.549619\text{E-}07(h)^2-4.046556\text{E-}13(h)^3]/144. \qquad (35)$$

In equations (30), (34), (35) h represents the altitude. For any altitudes greater than 150,000 ft. a pressure of 0 PSI is used in SSFAN for the value of PAMB.

3.3 Assumptions

Not applicable.

3.4 Computation

The logic involved in SSFAN concerns the selecting of the proper output subroutines chosen by the user. The desired output types, which are read in on the sixth data card, are stored in the PRINT array. The type output number is converted to an integer called IPRINT. A computed Go To statement will then select the proper output corresponding to the output type.

3.5 Approximations

The equations for viscosity-pressure correction and ambient pressure are approximations as any attempt to mathematically formulate a physical system would be. The pressure at altitude equations (30), (34), (35) are based on a constant gravitational value 32.2 ft/sec^2 for all altitudes. Air is also assumed to be a perfect gas. These approximations will yield slightly lower pressure values at high altitudes when compared with the standard atmosphere tables.

3.6 Limitations

Not applicable.

3.7 SSFAN Variable Names

Variable	Description	Dimensions
BLEG	General purpose array	--
BPS	Dynamic element junction storage array for sorting	--
BRANCHP	Dynamic element data storage array	--
CALC1	Matrix of coefficients	--
CALC2	NU matrix of constants	--
CONNECT	Static element data storage array	--
CONS	Static element junction storage array for sorting	--
DDENS	Array of weight density values input by user	LB/FT^3
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DTEMP	Array of temperature values corresponding to the DDENS array	°F
D100	Weight density at 100°F	LB/FT^3
EQD1	Average diameter of leg	IN
EQD2	Orifice diameter	IN
ERR	Error indicator	--
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
FTYPE	Array containing the system title and fluid name	--
I	Integer counter	--
ICENT	Array containing number of non-zero elements in each column of CALC1	--

3.7 (Continued)

Variable	Description	Dimensions
ICOL	Array containing column location of each non-zero element of CALC1	--
IDES	Storage array of element names	--
IDIAG	Array which identifies which columns of CALC1 contain positive conductance values	--
IERROR	Error indicator 0 = No program errors $\pm$ = Program termination	--
ILEP	Array of leg numbers with up and downstream pressure points	--
INEG	Array which stores the second appearance of a negative conductance value	--
IORDER	Array giving pivot selection based on min-row min-column criteria	--
IPQL2	Integer counter	--
IRENT	Array containing number of non-zero elements in each row of CALC1	--
ITER	Iteration count	--
ITY	Storage array of element types	--
ITYPE	Integer element type	--
J	Integer counter	--
JCENT	Array identifying the number of non-zero entries in each column of CALC1	--
JCOL	Array identifying the non-zero filled columns of CALC1; the rows correspond to the rows of CALC1; elements in each row correspond to column number in each row of CALC1	--
JEM.	Total number of pressure points in the system	--
JNEG	Array which identifies which column in CALC1 contains the first appearance of a negative conductance value in CALC1 array	--

[illegible] (Continued)

Variable	Description	Dimensions
JRENT	Array identifying the number of non-zero entries in each row of CALCI	--
J1	Element type indicator	--
J2	Element type indicator	--
J3	Element type indicator	--
K	Integer counter	--
M	Integer counter	--
ML	Total number of legs	--
N	Number of data cards for one element and number of fixed pressure points	--
NAB	Total number of floating branch points	--
NBP	Total number of elements in BRANCHP array	--
NBP2	Total length of BRANCHP array	--
NC	Total number of elements in CONNECT array	--
NCT	Integer indicator for number of data cards	--
NC2	Total length of CONNECT array	--
NJ3	Integer counter	--
NL	Total length of BLEG & ILEP array	--
NPQ	Total number of rows in PQL array	--
NPQL2	Array containing row location in BRANCHP array of element with fixed pressure	--
NVIS	Total number of viscosity data points input by user	--
N16	Length of QT16 array	--
N17	Length of QT17 array	--

3.7 (Continued)

Variable	Description	Dimensions
N18	Length of QT18 array	--
N19	Length of QT19 array	--
PAMB	Atmospheric ambient pressure	PSI
PQL	Array containing calculated pressures, element types and junction numbers	--
PQL2	Array of constant pressure point junction	--
PRINT	Array containing output types	--
QT15	Storage array for quasi-transient temperatures	--
QT16	Storage array for additional element which used in quasi-transient calculations	--
QT17	Storage array for quasi-transient valve data	--
QT18	Storage array for quasi-transienr valve data	--
TEMP	Fluid temperature	°F
TEMPF	Final fluid temperature	°F
TEMPINC	Fluid temperature increment	°F
TEMP14	Fluid temperature at branch point array	°F
TODAY	Current Julian date	--
TYPE	Element type	--
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
VTEMP	Temperature data values for viscosity input by user	°F
VVISC	Viscosity data values input by user	CENTISTOKES
V100	Fluid viscosity at 100°F	CENTISTOKES

3.8 MAIN PROGRAM SSFAN LISTING

```
      PROGRAM SSFAN(DATA,OUTPUT TAPE5=DATA,TAPE6=OUTPUT)
C              STEADY STATE FLOW ANALYSIS----SSFAN        22 FEB 1979
C              DATA INPUT - SORTING - STORAGE
      DIMENSION ITY(30),IDES(30),TYPE(19)
      DIMENSION BLANK1(1),BLANK2(1),BLANK3(1)
      DIMENSION TEMP14(100,2)
      DIMENSION QT15(3),QT16(10,18),QT17(5,18),QT18(5,18),QT19(108,10)
      DIMENSION CALC1(70,9),JCOL(70,5),CALC2(70),JRENT(70)
      DIMENSION JCENT(70),IDIAG(70),JNEG(70),INEG(70)
      DIMENSION IRENT(70),ICENT(70),IORDER(70,3),ICOL(70,9)
      COMMON TEMPF,TEMPINC
      COMMON AFBP(20),BRANCHP(70,22),CONNECT(100,8),ILEP(70,2)
      COMMON AFBPS(20),BPS(70,6),CONS(100,3),BLEG(70,16),PQL(70,8)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      COMMON /BLK2/VVISC(9),VTEMP(9),DDENS(2),DTEMP(2),NVIS
      COMMON /BLK3/NPQL2(20),PQL2(20)
      COMMON /BLK7/IERROR,ITER
      COMMON /BLK8/TODAY(1)
      COMMON /BLK9/FTYPE(16),PRINT
      DATA NBP2,NC2,NL,NPQ/70,100,70,70/,BCHK/'          '/
      DATA N16,N17,N18,N19/10,5,5,10/
      DATA ITY/1,2,3,4,5,6,7,8,9,10,11,12,13,21,22,23,24,
     125,31,32,33,34,35,36,37,38,91,92,100,100/
      DATA IDES/'TUBE','UNION','CHECK VLV','ACTUATOR',
     1'PUMP','FILTER','PP-QPT-ACC','MOTOR','RESERVOIR',
     2'SPECIAL','HOSE','HEAT EXCH','FLOAT BP','45 DEG EL',
     3'90 DEG EL','REDUCER','TEE','CROSS','1-WAY RSTR',
     4'2-WAY RSTR','RELIEF VLV','4W-3P VLV','3W-2P VLV',
     5'2W-2P VLV','FLOW REG','ORF SIZER','APPX RESV',
     6'CNST P RSV','**********','**********'/
      CALL DATE(TODAY)
      READ(5,7)(FTYPE(M),M=1,8)
      READ(5,7)(FTYPE(M),M=9,16)
      CALL OPUT4
      WRITE(6,2)
      WRITE(6,1)
    1 FORMAT(1X,9('0'),10('1'),10('2'),10('3'),10('4'),10('5'),10('6'),
     110('7'),'8')
    2 FORMAT(' ',33('*'),'COLUMN NUMBERS',33('*'))
      WRITE(6,3)
    3 FORMAT(' ',8('1234567890'))
      WRITE(6,4)
    4 FORMAT(' ',5(8('+'),8('$')))
      IERROR=0
      ERR=0.
      NJ3=0
      I16=0
      I17=0
      I18=0
      I19=1
      READ(5,12)NVIS,BLANK1(1),(VVISC(M),M=1,NVIS)
      READ(5,12)NVIS,BLANK2(1),(VTEMP(M),M=1,NVIS)
      READ(5,15)DDENS(1),DDENS(2),DTEMP(1),DTEMP(2)
      READ(5,15)TEMP,TEMPF,TEMPINC,ALT,PRINT
      WRITE(6,5)
    5 FORMAT(/' CARD 1 ',23('+'),'TITLE',45('+'))
      WRITE(6,8)(FTYPE(M),M=1,8)
      WRITE(6,6)
    6 FORMAT(/' CARD 2 ',23('+'),'FLUID',45('+'))
      WRITE(6,8)(FTYPE(M),M=9,16)
    7 FORMAT(8A10)
      WRITE(6,9)
    8 FORMAT(' ',8A10)
    9 FORMAT(/' CARDS 3&4 ',9('+'),'VISCOSITY-TEMPERATURE DATA',35('+'))
      WRITE(6,13)NVIS,BLANK1(1),(VVISC(M),M=1,NVIS)
      WRITE(6,13)NVIS,BLANK2(1),(VTEMP(M),M=1,NVIS)
      WRITE(6,10)
   10 FORMAT(/' CARD 5 ',14('+'),'DENSITY-TEMPERATURE DATA',35('+'))
      WRITE(6,11)DDENS(1),DDENS(2),DTEMP(1),DTEMP(2)
   11 FORMAT(' ',10F8.3)
   12 FORMAT(I1,A7,9F8.2)
   13 FORMAT(' ',I1,A7,9F8.2)
      WRITE(6,14)
   14 FORMAT(/' CARD 6 ',5('+'),'INITIAL TEMP-FINAL TEMP-TEMP INCR-ALTIT
     1UDE-PRINT-OPTIONS',12('+'))
      WRITE(6,11)TEMP,TEMPF,TEMPINC,ALT,PRINT
   15 FORMAT(10F8.3)
      WRITE(6,16)
   16 FORMAT(//11X,'TYPE',2X,'DESCRIPTION',4X,'TYPE',
     12X,'DESCRIPTION',4X,'TYPE',2X,'DESCRIPTION',
     2/11X,4('+'),2X,11('+'),4X,4('+'),2X,11('+'),
     34X,4('+'),2X,11('+'))
      DO 17 I=1,10
   17 WRITE(6,18)ITY(I),IDES(I),ITY(I+10),IDES(I+10),ITY(I+20),
     1IDES(I+20)
   18 FORMAT(12X,I2,3X,A10,6X,I2,3X,A10,6X,I2,3X,A10)
      WRITE(6,19)
   19 FORMAT('1CARDS 7 & ON ',27('+'),'ELEMENT DATA',28('+'))
      WRITE(6,2)
      WRITE(6,1)
      WRITE(6,3)
      WRITE(6,4)
   20 READ(5,21)N,BLANK3(1),(TYPE(M),M=1,9)
      NCT=1
   21 FORMAT(I1,A7,9F8.3)
      WRITE(6,22)N,BLANK3(1),(TYPE(M),M=1,9)
   22 FORMAT(' ',I1,A7,9F8.3)
      AN=TYPE(1)
      IF(AN.EQ.1..OR.AN.EQ.11.)CALL BLOSS(NCT,TYPE)
      IF(N.GE.2)READ(5,15)(TYPE(M),M=10,19)
      IF(N.GE.2)NCT=2
      IF(BLANK3(1).EQ.BCHK)GO TO 24
      ERR=ERR+1.
      WRITE(6,23)
   23 FORMAT(' ',5X,'***ERROR*** DATA IN COLUMNS 2 THRU 8')
   24 ITYPE=TYPE(1)
```

3.8 (Continued)

```
      IF(ITYPE.EQ.0)GO TO 58
      IF(ITYPE.GT.10)GO TO 25
      IF(ITYPE.GT.2)GO TO 35
      GO TO (37,37),ITYPE
   25 J1=ITYPE/10
      J2=ITYPE-J1*10
      GO TO (26,27,28,30,30,30,30,30,29),J1
   26 GO TO (37,37,52,32,41,41,41,41,41),J2
      GO TO 30
   27 GO TO (37,37,37,35,35),J2
      GO TO 30
   28 GO TO (35,37,35,35,35,35,35,35),J2
      GO TO 30
   29 GO TO (35,35),J2
   30 ERR=ERR+1.
      WRITE(6,31)
   31 FORMAT(' ',11X,'^**ERROR*** ILLEGAL ELEMENT TYPE')
      GO TO 56
   32 DO 33 J3=3,9
      IF(TYPE(J3).EQ.0.)GO TO 34
      NJ3=NJ3+1
      TEMP14(NJ3,2)=TYPE(2)
   33 TEMP14((NJ3),1)=TYPE(J3)
   34 GO TO 54
   35 NBP=NBP+1
      IF(N.EQ.2)WRITE(6,57)(TYPE(M),M=10,19)
      DO 36 J=1,17
   36 BRANCHP(NBP+(J-1)*NBP2)=TYPE(J)
      GO TO 54
   37 NC=NC+1
      IF(N.GE.2)WRITE(6,57)(TYPE(M),M=10,19)
      IF(N.GE.2)CALL BLOSS(NCT,TYPE)
      IF(N.EQ.NCT)GO TO 39
   38 READ(5,15)(TYPE(M),M=10,19)
      WRITE(6,57)(TYPE(M),M=10,19)
      CALL BLOSS(NCT,TYPE)
      NCT=NCT+1
      IF(NCT.EQ.N)GO TO 39
      GO TO 38
   39 DO 40 J=1,8
   40 CONNECT(NC+(J-1)*NC2)=TYPE(J)
      GO TO 54
   41 IF(N.EQ.2)WRITE(6,57)(TYPE(M),M=10,19)
      J3=ITYPE-14
      GO TO (42,44,46,48,50),J3
   42 DO 43 I=1,3
      J=I+1
   43 QT15(I)=TYPE(J)
      GO TO 54
   44 I16=I16+1
      DO 45 I=1,18
      J=I+1
   45 QT16(I16,I)=TYPE(J)
```

```
      GO TO 54
   46 I17=I17+1
      DO 47 I=1,18
      J=I+1
   47 QT17(I17,I)=TYPE(J)
      GO TO 54
   48 I18=I18+1
      J=I+1
   49 QT18(I18,I)=TYPE(J)
      GO TO 54
   50 I19=I19+1
      DO 51 I=1,5
      J=I+1
   51 QT19(I,I19)=TYPE(J)
      GO TO 54
   52 DO 53 J=2,9
      IF(TYPE(J).EQ.0.)GO TO 54
      NAB=NAB+1
   53 AFBP(NAB)=TYPE(J)
   54 DO 55 M=1,19
   55 TYPE(M)=0.
      GO TO 20
   56 IF(N.EQ.2)WRITE(6,57)(TYPE(M),M=10,19)
      GO TO 20
   57 FORMAT(' ',10F8.3)
   58 CONTINUE
      CALL VISD
      IF(ALT.LE.36089.)PAMB=(1.6676*(1.-ALT/145378.))**5.256
      IF(ALT.GT.36089.AND.ALT.LE.65E3)PAMB=3.2825
     1 *EXP(4.806E-5*(36089.-ALT))
      IF(ALT.GT.65E3.AND.ALT.LE.15E4)PAMB=(867.42375-1.990498E-2*ALT
     1 +1.549619E-7*(ALT**2)-4.046556E-13*(ALT**3))/144.
      IF(ALT.GT.15E4)PAMB=.0197361
      CALL SDSORT(AFBP,BRANCHP,CONNECT,AFBPS,BPS,CONS,NBP2,NC2)
      IF(IERROR.GT.0)GO TO 76
      CALL BUILD(ILEP,CONS,BPS,AFBPS,BLEG,PQL,NBP2,NC2,NL,NPQ,
     1BRANCHP,CONNECT,ML)
      DO 59 I=1,NPQ
   59 PQL(I+7*NPQ)=TEMP
      DO 61 I=1,NPQ
      IF(PQL(I).EQ.0.)GO TO 62
      DO 60 J=1,100
      IF(TEMP14(J,1).EQ.0.)GO TO 61
      DO 60 K=3,6
      IF(TEMP14(J,1) EQ.PQL(I+K*NPQ))PQL(I+7*NPQ)=TEMP14(J,2)
   60 CONTINUE
   61 CONTINUE
   62 CONTINUE
      DO 64 I=1,NL
      IF(ILEP(I).EQ.0.)GO TO 65
      BLEG(I+2*NL)=10.
      IF(BLEG(I+6*NL).EQ.0.)GO TO 63
      EQD1=BLEG(I+3*NL)**4
```

3.8 (Continued)

```
      EQD2=BLEG(I+6*NL)**4
      BLEG(I+3*NL)=((EQD1*EQD2)/(EQD1+EQD2))**.25
      BLEG(I+6*NL)=0.
   63 BLEG(I+7*NL)=.0001
      BLEG(I+4*NL)=0.
      IF(BLEG(I+3*NL).EQ.0.)BLEG(I+12*NL)=2.
      IF(BLEG(I+3*NL).EQ.0.)BLEG(I+3*NL)=1.
      BLEG(I+13*NL)=(PQL(ILEP(I)+7*NPQ)+PQL(ILEP(I+NL)+7*NPQ))/2.
      TEMP=BLEG(I+13*NL)
      VISC=0.
      CALL VISD
      FLUIDK=(FLUIDF**(560./(460.+TEMP)))*.00023
      BLEG(I+14*NL)=FLUIDK
   64 BLEG(I+15*NL)=VISC
   65 N=0
      JEM=0
      IPQL2=0
      DO 66 I=1,NBP2
      IF(PQL(I).EQ.0.)GO TO 67
      JEM=JEM+1
      IF(PQL(I).EQ.-1.)GO TO 66
      N=N+1
      IPQL2=IPQL2+1
      NPQL2(IPQL2)=I
      PQL2(IPQL2)=PQL(I)
   66 CONTINUE
   67 CONTINUE
      IF(PRINT.EQ.1.)GO TO 68
      IF(PRINT.EQ.2.)GO TO 69
      IF(PRINT.EQ.3.)GO TO 70
      GO TO 71
   68 WRITE(6,72)(I,(ILEP(I,J),J=1,2),I=1,ML)
   69 WRITE(6,73)((PQL(I,J),J=1,8),I=1,JEM)
   70 WRITE(6,74)((BLEG(I,J),J=1,16),I=1,ML)
   71 CONTINUE
   72 FORMAT(3I10)
   73 FORMAT(8F12.3)
   74 FORMAT(8F14.4)
      CALL OPUT4
      CALL OPUT3(ILEP,BLEG,NL,PQL,NPQ)
      IF((QT15(2)-QT15(1)).LE.0.)GO TO 75
      CALL QTCALC(BRANCHP,NBP2,PQL,NPQ,BLEG,ILEP,NL,QT16,N16,
     1QT17,N17,QT18,N18,QT19,N19,ML,N,JEM,CALC1,JCOL,CALC2,
     2JRENT,JCENT,IDIAG,JNEG,INEG,IRENT,ICENT,IORDER,ICOL,PQL2,NPQL2)
      GO TO 76
   75 CALL CALC(ML,N,JEM,BLEG,PQL,CALC1,JCOL,CALC2,JRENT,JCENT,
     1IDIAG,JNEG,INEG,BRANCHP,NBP2,NL,ILEP,IRENT,ICENT,IORDER,ICOL,
     2NPQ)
      CALL OPUT4
      CALL OPUT2(ILEP,BLEG,NL,PQL,NPQ)
   76 CONTINUE
      END
```

SECTION IV

DATA INPUT

The Input section of the Main Program is designed to minimize user time and effort in the preparation of the data deck. All data cards are divided into 10 columns of 8 characters each, with only the title card and fluid name card not following this format.

The first six cards of the data deck contain the system parameters. They along with the last data card are the only cards that must be inserted in a specific order. The system parameters include the system title, fluid name, viscosity-temperature data points, density-temperature data points, system operating temperature, altitude and the type of data output. The remainder of the deck contains the system element data. The description of how the element physical data is entered on the cards along with any other information concerning the data deck setup may be found in the SSFAN Users Manual (AFAPL-TR-76-43, Vol. V).

The Input has no restriction as to the placing of element data cards in the data deck. Each element card or cards depending on the type of element, may be inserted in any order. If new elements are added to the system, the data cards containing these elements may be located anywhere the user finds it best to insert them. The Input section of the program processes each element individually, therefore a rigid format for element order in the data deck need not be followed.

The final card for data input has a zero element type. The reading of a zero element type will terminate the data processing.

Figure 10 presents a simple flow chart for the input processing of the data deck. The SSFAN Main Program reads in the data cards and writes out the data exactly as it was read in. The data is processed one card at a time for system parameter cards, and one or more cards for element data. The data is stored in the appropriate array as it is read in. If the data being processed is a tube or hose, its energy loss coefficient is computed and stored before another card is read. When an illegal format occurs on a data card an error message is printed out and the processing continues with the next card. The card with element type zero causes the main program to stop reading data cards.

Figure 11 presents a summary of data card input parameters for all elements.

Refer to Figure 12 for examples of data cards written out exactly as they were read in.

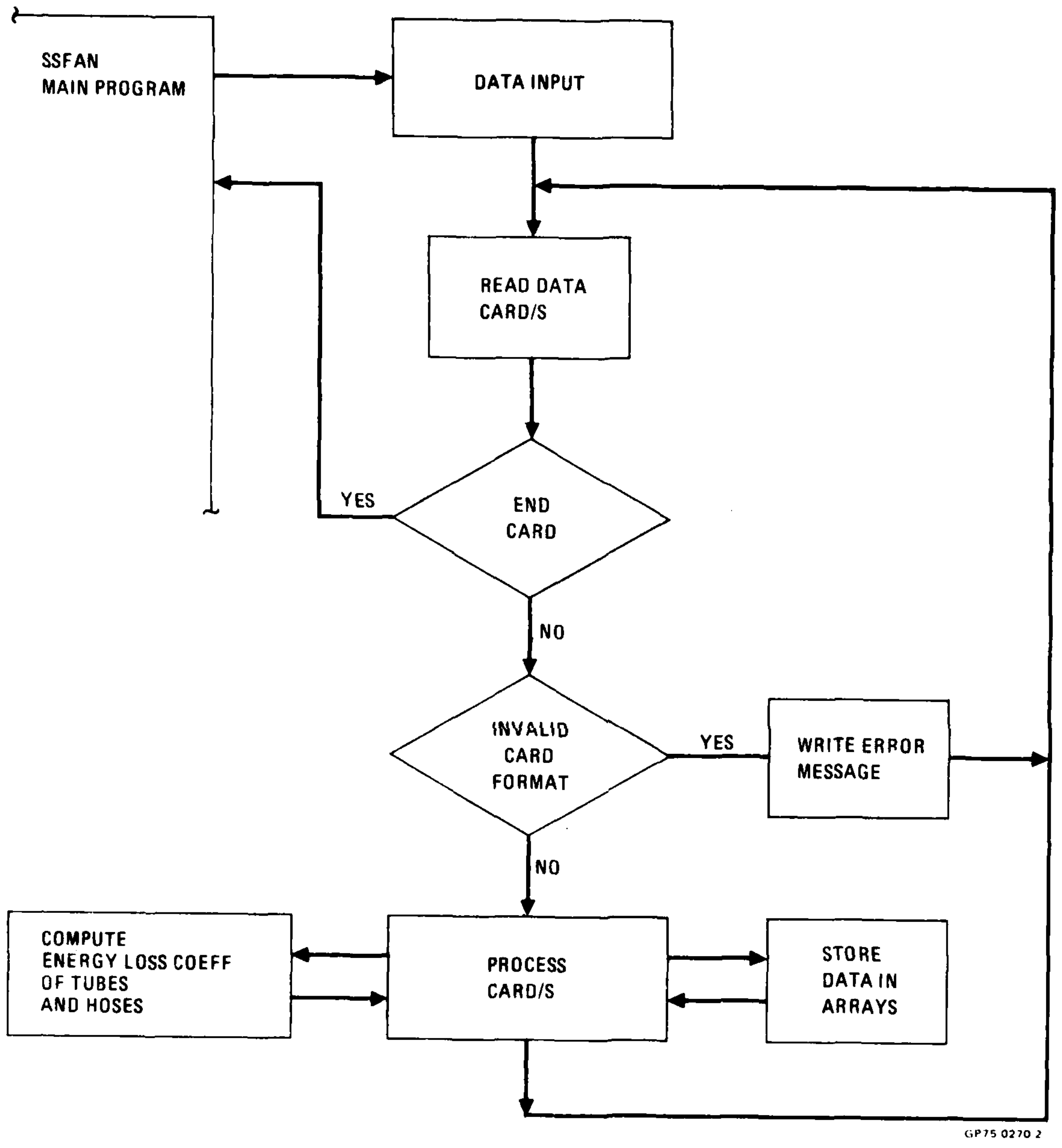


FIGURE 10
GENERALIZED DATA INPUT FLOW CHART

Element Name	Elem Type	CARD 1										Card 2									
		1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	
Tube (Line)	1	No. Cards 1 or More	Elem Type	JCT 1	JCT 2	Size	Wall Th.	Len	Bend	Bend	Bend	Bend	Bend	Bend	Bend	Bend	Bend	Bend	Bend	Bend	Bend
Union	2	No. Cards 1	Elem Type	JCT 1	JCT 2	Size 1	Size 2														
Check Valve	3	No. Cards 1	Elem Type	JCT 1	JCT 2	Size 1	Size 2	CR Press													
Simple Actuator	4	No. Cards 2	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Ext Area	Retr Area	Seal Frict	External Load	Stroke	Piston Pos	Piston Dia	Vlv Jct to Ext Act.						
Pump-Var Del	5	No. Cards 2	Elem Type	JCT 1	JCT 2	JCT 3	Size 1	Size 2	Size 3	Actual Pump RPM	Rated Pump RPM	Rated Flow	Rated Press 0 Flow	Rated Press Full Flow	PS Min	RCHP	RCDL	PSET			
Filter	6	No. Cards 2	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Internal Fluid Vol	Rated Q Clean Elem	Rated ΔP Clean Elem	VISC at Rated Conds	Contam Factor	Relief Crack Press	Bypass ΔP at Rated Q							
QPT-PPT-ACC	7	No. Cards 1	Elem Type	JCT	Size	Type of QPT-PPT-ACC	Press or Flow	Initial Press	ΔVol												
Motor	8	No. Cards 2	Elem Type	JCT 1 (In)	JCT 2 (Out)	JCT 3 (CD) (Opt)	Size 1	Size 2	Size 3	Displacement	Load Torque	Eff.	CD Leakage								
Bootstrap RESV	9	No. Cards 2	Elem Type	JCT1(R)	JCT2(BS)	JCT3(S)	Size 1	Size 2	Size 3	Hi Press Area	Lo Press Area	Seal Frict									
Special	10	No. Cards 1 or 2	Elem Type	JCT 1	JCT 2	Size 1	Size 2	VISC	No. Data Points	Press	Drop 1	thru 6 Flow 1 thru 6									
Hose	11	No. Cards 1	Elem Type	JCT 1	JCT 2	Size	Hose ID	Len	Bend	Bend	Bend										
Heat Exch	12	No. Cards 1	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Rated Q	Rated ΔP	VISC											
Float BP	13	No. Cards 1	Elem Type	JCT 1																	
JCT PT Temp Input	14	No. Cards 1	Elem Type	Temp	JCT	JCT	JCT	JCT	JCT	JCT	JCT										
Time Interval and Step	15	No. Cards 1	Elem Type	Initial Time	Final Time	Time Step (Opt)															
Add. Data for Actuator	16	No. Cards 1 or 2	Elem Type	Act. Elem Type	Act. Extend JCT No.	No. of Load vs Strk Pts	Piston Pos. 1	Piston Pos. 2	Piston Pos. 3	Piston Pos. 4	Piston Pos. 5	Piston Pos. 6	Piston Pos. 7	Act. Load 1	Act. Load 2	Act. Load 3	Act. Load 4	Act. Load 5	Act Load 6	Act. Load 7	
Add. Data for Dyn. Accum.	16	No. Cards 1	Elem Type	Dyn. Accum Elem Typ	Accum Port JCT No.	Initial Fluid Vol.	Init Gas/Fld Pass Area														
	16																				
Add. Data for Hyd Motor	16	No. Cards 1 or 2	Elem Type	Hyd Motor Elem Typ	Mtr Inlt Press Prt JCT No.	PSID for Eff vs RPM	No. of Eff vs RPM Prs	RPM 1	RPM 2	RPM 3	RPM 4	RPM 5	RPM 6	RPM 7	EFF 1	EFF 2	EFF 3	EFF 4	EFF 5	EFF 6	EFF
Pos. vs Time	17	No. Cards 1 or 2	Elem Type	Valve Elem. Type	Vlv Inlt Press Prt JCT No.	No. VLV Pos vs Time Prs	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6	Time 7	Valve Pos 1	Valve Pos 2	Valve Pos 3	Valve Pos 4	Valve Pos 5	Valve Pos 6	Valve Pos 7	

FIGURE 11 Summary of Element Data Card Input

Pos. vs. Pos. Data	18	No. Cards 2	Elem Type	Act. Elem Type	Act. Extd Prt JCT No.	Valve Elem Type	Vlv Inlt Press JCT No.	No. of Pos. vs Pos. Pts	Act. Stroke Pos 1	Act. Stroke Pos 2	Act. Stroke Pos 3	Act. Stroke Pos 4	Act. Stroke Pos 5	Act. Stroke Pos 6	Valve Pos 1	Valve Pos 2	Valve Pos 3	Valve Pos 4	Valve Pos 5	Valve Pos 6	
Output Data	19	No. Cards	Elem Type																		
45° Elbow	21	No. Cards 1	Elem Type	JCT 1	JCT 2	Size 1	Size 2														
90° Elbow	22	No. Cards 1	Elem Type	JCT 1	JCT 2	Size 1	Size 2														
Reducer	23	No. Cards 1	Elem Type	JCT 1	JCT 2	Size 1	Size 2														
Tee	24	No. Cards 1	Elem Type	JCT 1	JCT 2	JCT 3	Size 1	Size 2	Size 3												
Cross	25	No. Cards 1	Elem Type	JCT 1	JCT 2	JCT 3	JCT 4	Size 1	Size 2	Size 3	Size 4										
1 Way Restrictor	31	No. Cards 2	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Orf Dia or Rated ΔP	Dischg Coeff or Rated Q	FF Crack Press (Opt)	Free Flo Rated P (Opt)	Free Flo Rated Q (Opt)									
2 Way Restrictor	32	No. Cards 1	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Orf Dia or Rated ΔP	Dischg Coeff or Rated Q												
Relief Valve	33	No. Cards 1	Elem Type	JCT 1	JCT ?	Size 1	Size 2	Relief Press Setting													
4W-3P Valve	34	No. Cards 2	Elem Type	JCT 1	JCT 2	JCT 3	JCT 4	Size 1	Size 2	Size 3	Size 4	Rated Q 1 to 3 or 4	Rated ΔP	Visc.	Leak Q 1 to 3	ΔP for Leak Q	Oper Cntl Code				
3W-2P Valve	35	No. Cards 2	Elem Type	JCT 1	JCT 2	JCT 3		Size 1	Size 2	Size 3		Rated Q 1 to 3	Rated ΔP	Visc.	Leak Q 1 to 3	ΔP for Leak Q	Oper Cntl Code				
2W-2P Valve	36	No. Cards 2	Elem Type	JCT 1	JCT 2			Size 1	Size 2			Rated Q 1 to 2	Rated ΔP	Visc.	Leak Q 1 to 2	ΔP for Leak Q	Oper Cntl Code				
Flow Regulator	37	No. Cards 2	Elem Type	JCT 1	JCT 2			Size 1	Size 2			Rated Q	Min ΔP	Visc.							
Restrictor Sizer	38	No. Cards 2	Elem Type	JCT 1	JCT 2			Size 1	Size 2												
Appendix RESV	91	No. Cards 1	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Hi Press Area	Lo Press Area	Seal Frict											
Constant Press RESV	92	No. Cards 1	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Press													

FIGURE 11 (Continued)

```
********************************************************************************
**********************************SSFAN PROGRAM**********************************
********************************************************************************

DATA DECK---------------SSFAN  --  SAMPLE CASE NUMBER 1

*******************************COLUMN NUMBERS************************************
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
++++++++++$$$$$$$$$$++++++++++$$$$$$$$$$++++++++++$$$$$$$$$$++++++++++$$$$$$$$$$

CARD 1 +++++++++++++++++++++++TITLE++++++++++++++++++++++++++++++++++++++++++++++
SSFAN  --  SAMPLE CASE NUMBER 1

CARD 2 +++++++++++++++++++++++FLUID++++++++++++++++++++++++++++++++++++++++++++++
MIL-H-5606A

CARDS 3&4 ++++++++VISCOSITY-TEMPERATURE DATA++++++++++++++++++++++++++++++++++++
9        2000.00   345.00  130.00    36.00    14.50     7.70     4.80     3.40     2.40
9         -65.00   -40.00    0.00    50.00   100.00   150.00   200.00   250.00   300.00

CARD 5 ++++++++++++++DENSITY-TEMPERATURE DATA++++++++++++++++++++++++++++++++++++
  56.200  43.700 -65.000 275.000

CARD 6 +++++INITIAL TEMP-FINAL TEMP-TEMP INCR-ALTITUDE-PRINT-OPTIONS++++++++++++
 100.000 100.000   0.000   0.000   1.000

       TYPE  DESCRIPTION      TYPE  DESCRIPTION      TYPE  DESCRIPTION
       ++++  +++++++++++      ++++  +++++++++++      ++++  +++++++++++
         1   TUBE              11   HOSE              33   RELIEF VLV
         2   UNION             12   HEAT EXCH         34   4W-3P VLV
         3   CHECK VLV         13   FLOAT RR          35   3W-2P VLV
         4   ACTUATOR          21   45 DEG EL         36   2W-2P VLV
         5   PUMP              22   90 DEG EL         37   FLOW REG
         6   FILTER            23   REDUCER           38   ORF SIZER
         7   RP-OPT-ACC        24   TEE               91   APPX RESV
         8   MOTOR             25   CROSS             92   CNST P RSV
         9   RESERVOIR         31   1-WAY RSTR        **   **********
        10   SPECIAL           32   2-WAY RSTR        **   **********
```

FIGURE 12   Example of Data Cards Printout

```
CARDS 7 & 8 ++++++++++++++++++++++++++++ELEMENT DATA++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++COLUMN NUMBER++++++++++++++++++++++++++++++++++++++
0000000001111111111222222222233333333334444444444555555555566666666667777777777
1234567890123456789012345678901234567890123456789012345678901234567890123456789
+++++++++$$$$$$$$$$++++++++++$$$$$$$$$$++++++++++$$$$$$$$$$++++++++++$$$$$$$$$$
2             4.000   145.000   135.000     4.000     4.000     4.500     3.250    35.0001000.000
   5.000      2.500     2.000    95.000     0.000     0.000     0.000     0.000     0.000     0.000
2             4.000   130.000   170.000     4.000     4.000     3.650     2.900    25.0001000.000
   5.000      2.500     2.000    95.000     0.000     0.000     0.000     0.000     0.000     0.000
2             1.000    40.000    50.000     5.000      .065   142.500    30.000    45.000   120.000
  65.000     23.000    72.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
1             1.000   230.000   235.000    10.000      .028    63.200    42.000    33.200     0.000
2            34.000    75.000    90.000    25.000   100.000     6.000     8.000     4.000     4.000
  25.000    200.000    14.000      .1003000.000     1.000     0.000     0.000     0.000     0.000
2             2.000   250.000   260.000   265.000    10.000     4.000    16.000     1.655    95.300
   2.000      0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
2             1.000   205.000   210.000     5.000      .028   147.300    23.000    38.000    90.000
  83.000    121.000   150.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
2             5.000     5.000    10.000    15.000    15.000    12.000     4.0003750.0003750.000
  50.0003000.0002750.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
1             2.000   245.000   250.000    10.000    10.000     0.000     0.000     0.000     0.000
1             2.000   235.000   240.000    10.000    10.000     0.000     0.000     0.000     0.000
1             2.000   220.000   215.000     4.000     4.000     0.000     0.000     0.000     0.000
1             1.000   225.000   220.000     4.000      .020    75.200    57.000    52.000    90.000
1             2.000    15.000   225.000     4.000     4.000     0.000     0.000     0.000     0.000
1             2.000   200.000   205.000     4.000     4.000     0.000     0.000     0.000     0.000
1             1.000   170.000   200.000     4.000      .028   128.500    97.000    31.000    68.000
1             1.000   165.000   160.000     4.000      .020   123.000    95.000    77.000    22.000
1             2.000   170.000   165.000     4.000     4.000     0.000     0.000     0.000     0.000
1             2.000   150.000   100.000     4.000     4.000     0.000     0.000     0.000     0.000
1             1.000   155.000   150.000     4.000      .020    95.600   147.000    23.000    58.000
1            24.000   175.000   160.000   155.000     4.000     4.000     4.000     0.000     0.000
1             1.000   170.000   175.000     4.000      .020    75.200   120.000 [illegible]     0.000
1             2.000   [illegible]   [illegible]     4.000     4.000
                                145.000     4.000

                              [illegible]    15.000   [illegible]                                   0.000
1             1.000   [illegible]   [illegible]    16.000      .042    95.000           [illegible]    25.000
1             2.000   275.000   250.000    15.000    15.000     0.000     0.000     0.000     0.000
1            21.000    22.000    30.000    12.000    12.000     0.000     0.000     0.000     0.000
1             2.000   235.000     5.000    16.000    16.000     0.000     0.000     0.000     0.000
1             7.000   235.000     4.000     1.000    50.000   100.000     4.000     6.800     0.000
1             0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
```

FIGURE 12 (Continued)

4.1 MAIN PROGRAM DATA INPUT

The main program data input section reads the data from the data cards which describe the system parameters and elements. The data is stored in labeled and unlabeled common and other arrays for use by the main program and the individual subroutines. All input is made using an 80 column card field as the basic format. The program reads one or more cards as required by the system element models. Any element type that is not listed in the SSFAN users manual as being a valid element will be rejected by the program which will print out an error message.

If the element is a tube or a hose with bends, the energy loss coefficient is calculated by BLOSS subroutine and stored with the element data. The bend angles are then discarded.

[illegible].1.2 Description of Data Input

The data input section has two basic tasks to perform. A flow diagram of this data processing is presented in Figure 13. The first task involves the reading in of the system parameters. A separate read statement is used to input each of these cards. The first two cards are the title and fluid name cards to be used wherever the system title or fluid name must appear in the program. The title card data is read into a labeled common (BLK9) whose array name is FTYPE. Cards three and four contain the viscosity-temperature data for the hydraulic fluid to be used in the system. This data is stored in BLK2 under array names VVISC and VTEMP. Also stored in BLK2 is the density-temperature data contained on card five. The final value stored in BLK2 is the number of viscosity data points which are used later in the viscosity interpolation subroutine VISD. This value is read off the viscosity data card. The last or sixth card of the system parameters contains the system initial and final temperature and temperature increment (for multiple temperature runs), altitude and type of output requested. The temperature and altitude are stored in BLK1 and the output types are inserted into BLK9 in the PRINT array.

The second task sorts the system elements into their respective arrays. One element is processed at a time. The read statement will input as many data cards that are needed to fully describe a system element. This information is temporarily put into a buffer array named TYPE. After the element type is converted to an integer (ITYPE), a computed Go To corresponds the type number with a statement number. A do loop at each statement number empties the buffer array into the proper element array.

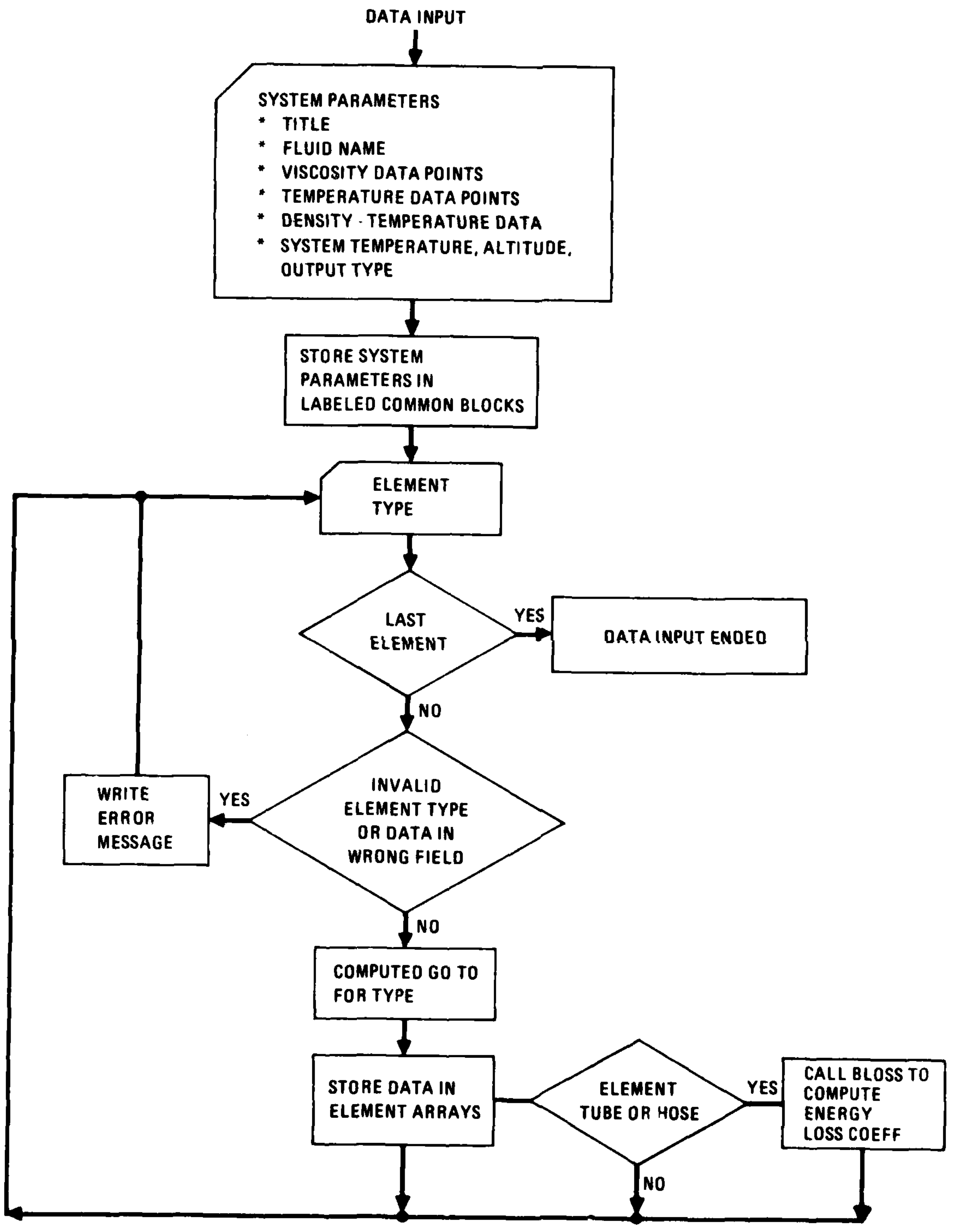

DATA INPUT
SYSTEM PARAMETERS
* TITLE
* FLUID NAME
* VISCOSITY DATA POINTS
* TEMPERATURE DATA POINTS
* DENSITY - TEMPERATURE DATA
* SYSTEM TEMPERATURE, ALTITUDE, OUTPUT TYPE
STORE SYSTEM PARAMETERS IN LABELED COMMON BLOCKS
ELEMENT TYPE
LAST ELEMENT
YES
DATA INPUT ENDED
NO
INVALID ELEMENT TYPE OR DATA IN WRONG FIELD
YES
WRITE ERROR MESSAGE
NO
COMPUTED GO TO FOR TYPE
STORE DATA IN ELEMENT ARRAYS
ELEMENT TUBE OR HOSE
YES
CALL BLOSS TO COMPUTE ENERGY LOSS COEFF
NO
GP75 0270 1

FIGURE 13
DATA INPUT FLOW DIAGRAM

All the basic input data are stored in AFBP, BRANCHP or CONNECT arrays. CONNECT array contains all the static elements (elements which have resistance coefficients calculated and stored and are not recalculated during the iterative calculations). BRANCHP array contains all dynamic elements (elements whose resistance coefficients are recalculated each iteration). AFBP array contains junction numbers input by the user to have pressures calculated which otherwise would not be calculated. Variable temperatures at branch points are stored in TEMP14 array. Quasi-transient calculation data are stored in arrays QT15, QT16, QT17, QT18 and QT19.

Figures 14 and 15 are summaries of data contained in CONNECT and BRANCHP arrays respectively. Figures 16 , 17 and 18 are actual computer printouts of data contained in CONNECT, AFBP and BRANCHP arrays, respectively, from SSFAN Sample Case Number 1.

4.1.2 Computations

BLOSS subroutine calculates energy loss coefficients for tubes and hoses. Figure 19 is the flow diagram for BLOSS. See Appendix C for detailed calculation method.

The DO loop parameters are set depending on the number of cards being read. The ratio of bend radius to inside diameter is set to 3.X (OD/ID) for tubes and 8. for hoses. To change either value requires changing the number on the appropriate card in BLOSS subroutine. Each bend angle is read from the TYPE data array one at a time, energy loss coefficient calculated and summed into a temporary storage location (ECOEF). When all the bend angles on one card are processed, ECOEF is summed into column 7 of the tube or hose data, replacing the first bend angle that was input.

Element Name	Elem Type	1	2	3	4	5	6	7	8
Tube (Line)	1	Elem Type	JCT 1	JCT 2	Size	Wall Th	Len	Energy Loss Coeff	
Union	2	Elem Type	JCT 1	JCT 2	Size 1	Size 2			
Hose	11	Elem Type	JCT 1	JCT 2	Size	Hose ID	Len	Energy Loss Coeff	
Heat Exch	12	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Rated Q	Rated ΔP	VISC
45° Elbow	21	Elem Type	JCT 1	JCT 2	Size 1	Size 2			
90° Elbow	22	Elem Type	JCT 1	JCT 2	Size 1	Size 2			
Reducer	23	Elem Type	JCT 1	JCT 2	Size 1	Size 2			
2-Way Restrictor	32	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Orf Dia or Rated P	Dischg Coeff or Rated Q	

FIGURE 14 CONNECT Array - Data Summary

Element Name	Elem Type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Check Valve	3	Elem Type	JCT 1	JCT 2	Size 1	Size 2	CR Press	Assv Dir 1 FF 2 CHKD	-1 From VCHFK IND FF													
Simple Actuator	4	Elem Type	JCT 1	JCT 2	PP JCT 1	PP JCT 2	Extend Area	RETR Area	Seal Friction	External Load	Stroke	Piston Pos	Piston Dia	Vlv. Jct to Ext Act.								
Pump - Var Del	5	Elem Type	JCT 1(S)	JCT2(P)	JCT 3 (CD)	PP JCT 1	PP JCT 2	PP JCT 3	Actual Pump RPM	Rated Pump RPM	Rated Flow	Rated Press 0 Flow	Rated Press Full Flow	PS Min	RCDP	RCDL	PSET					
Filter	6	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Initial Fluid Volume	Rated Q Clean Elem	RatedΔP Clean Elem	VISC at Rated [illegible]	Contam Factor	Relief Crack Press	Bypass ΔP at Rated Q									
QPT-PPT-ACC	7	Elem Type	JCT	PP JCT 1	Type of QPT-PPT-ACC	Press or Flow	Initial Press	Δ Vol														
Motor	8	Elem Type	JCT 1 (In)	JCT 2 (Out)	JCT 3 (CD) (OPT)	PP JCT 1	PP JCT 2	PP JCT 3	Displacement	Load Torque	Eff.	CD Leakage										
Bootstrap RESV	9	Elem Type	JCT1(R)	JCT2(BS)	JCT3(S)	PP JCT 1	PP JCT 2	PP JCT 3	HI Press Area	LO Press Area	Seal Friction	(R) Leg No.	(BS) Leg No.									
Special	10	Elem Type	JCT 1	JCT 2	Size 1	Size 2	VISC	No. Data Points	Press	Drop 1 Flow 1	thru 6 thru 6											
Tee	24	Elem Type	JCT 1	JCT 2	JCT 3	Size 1	Size 2	Size 3	JCT1 Leg No.	JCT2 Leg No.	JCT3 Leg No.	JCT1 MRK 1 Beg 0 End	JCT2 MRK 1 Beg 0 End	JCT3 MRK 1 Beg 0 End								
Cross	25	Elem Type	JCT 1	JCT 2	JCT 3	JCT 4	Size 1	Size 2	Size 3	Size 4	JCT1 Leg No.	JCT2 Leg No.	JCT3 Leg No.	JCT4 Leg No.	JCT1 MRK 1 Beg 0 End	JCT2 MRK 1 Beg 0 End	JCT3 MRK 1 Beg 0 End	JCT4 MRK 1 Beg 0 End				
1 Way Restrictor	31	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Orf Dia or RatedΔP	Dischg Coeff or Rated Q	FF Crack Press (Opt)	FF Rated ΔP (Opt)	FF Rated Q (Opt)	Assv Ind 1 Restr 2 FF		-1 From VCHEX Ind FF								
Relief Valve	33	Elem Type	JCT 1	JCT 2	Size 1	Size 2	Relief Press Setting	Assv Ind 1 FF 2 REV DIR	-1 From VCHECK Ind FF													
4W-3P Valve	34	Elem Type	JCT 1	JCT 2	JCT 3	JCT 4	PP JCT 1	PP JCT 2	PP JCT3	PP JCT4	Rated Q 1 to 3 or 4	Rated ΔP	VISC	Leak Q 1 to 3	ΔP for Leak Q	Oper Cntl Code	Leg No. JCT1 - JCT2 P-R	Leg No. JCT1 - JCT3P-C1	Leg No. JCT3 - JCT2 C1-R	Leg No. JCT1 - JCT4 P-C2	Leg No. JCT 4 - JCT2 C2-R	
3W-3P Valve	35	Elem Type	JCT 1	JCT 2	JCT 3	PP JCT1	PP JCT2	PP JCT3	Size 3		Rated Q 1 to 3	Rated ΔP	VISC	Leak Q 1 to 3	ΔP for Leak Q	Oper Cntl Code	Leg No. JCT 1 - JCT2 P-R	Leg No. JCT1 - JCT3 P-C1	Leg No. JCT3 - JCT2 C1-R	Leg No. JCT1 - JCT4 P-C2	Leg No. JCT4 - JCT2 C2-R	
2W-2P Valve	36	Elem Type	JCT 1	JCT 2	PP JCT 1	PP JCT 2	Size 1	Size 2			Rated Q 1 to 2	Rated ΔP	VISC	Leak Q 1 to 2	ΔP for Leak Q	Oper Cntl Code	Leg No. JCT1 - JCT2 P-R					
Flow Regulator	37	Elem Type	JCT 1	JCT 2	PP JCT 1	PP JCT 2	Size 1	Size 2		Used for Ind for Flochek	Rated Q	Min ΔP	VISC									
Restrictor Size	38	Elem Type	JCT 1	JCT 2	PP JCT 1	PP JCT 2	Size 1	Size 2														
Appendix Resv	91	Elem Type	JCT 1	JCT 2	PP JCT 1	PP JCT 2	HI Press Area	LO Press Area	Seal Friction			(R/S) Leg No.	(BS) Leg No.									
Constant Press Resv	92	Elem Type	JCT 1	JCT 2	PP JCT 1	PP JCT 2	Press															

FIGURE 15 BRANCHP Array - Data Summary

1.000	40.000	50.000	8.000	.065	142.500	.445	45.000
1.000	230.000	235.000	10.000	.023	63.200	.124	33.200
1.000	205.000	210.000	8.000	.023	147.300	.606	38.000
2.000	245.000	250.000	10.000	10.000	-0.000	-0.000	-0.000
2.000	235.000	240.000	10.000	10.000	-0.000	-0.000	-0.000
2.000	220.000	215.000	4.000	4.000	-0.000	-0.000	-0.000
1.000	225.000	220.000	4.000	.020	73.200	.337	59.000
2.000	15.000	225.000	4.000	4.000	-0.000	-0.000	-0.000
2.000	200.000	205.000	8.000	8.000	-0.000	-0.000	-0.000
1.000	190.000	200.000	8.000	.028	128.500	.267	31.000
1.000	165.000	160.000	4.000	.020	123.000	.253	77.000
2.000	170.000	165.000	4.000	4.000	-0.000	-0.000	-0.000
2.000	150.000	100.000	4.000	4.000	-0.000	-0.000	-0.000
1.000	155.000	150.000	4.000	.020	85.600	.265	23.000
1.000	130.000	175.000	4.000	.020	75.300	.225	71.300
2.000	185.000	180.000	4.000	4.000	-0.000	-0.000	-0.000
2.000	140.000	145.000	4.000	4.000	-0.000	-0.000	-0.000
1.000	135.000	140.000	4.000	.020	95.700	.206	77.000
2.000	125.000	130.000	4.000	4.000	-0.000	-0.000	-0.000
1.000	120.000	125.000	4.000	.020	30.000	.090	-0.000
1.000	110.000	115.000	4.000	.020	75.000	.206	23.000
2.000	95.000	110.000	4.000	4.000	-0.000	-0.000	-0.000
2.000	70.000	75.000	6.000	6.000	-0.000	-0.000	-0.000
1.000	75.000	80.000	6.000	.022	120.000	0.000	-0.000
1.000	60.000	70.000	6.000	.022	170.000	.138	15.000
2.000	290.000	295.000	4.000	4.000	-0.000	-0.000	-0.000
1.000	65.000	290.000	4.000	.020	26.500	.092	25.000
1.000	30.000	35.000	12.000	.042	85.300	.223	30.000
11.000	20.000	25.000	12.000	.090	57.000	.057	25.000
21.000	80.000	85.000	6.000	6.000	-0.000	-0.000	-0.000
1.000	245.000	255.000	4.000	.020	85.300	.213	33.700
2.000	255.000	260.000	4.000	4.000	-0.000	-0.000	-0.000
11.000	280.000	265.000	16.000	.875	25.000	.028	-0.000
22.000	10.000	20.000	12.000	12.000	-0.000	-0.000	-0.000
22.000	265.000	270.000	15.000	16.000	-0.000	-0.000	-0.000
1.000	270.000	275.000	16.000	.042	85.000	.144	31.000
2.000	275.000	280.000	16.000	16.000	-0.000	-0.000	-0.000
21.000	25.000	30.000	12.000	12.000	-0.000	-0.000	-0.000
2.000	235.000	5.000	16.000	16.000	-0.000	-0.000	-0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

FIGURE 16 CONNECT Array - Example Data

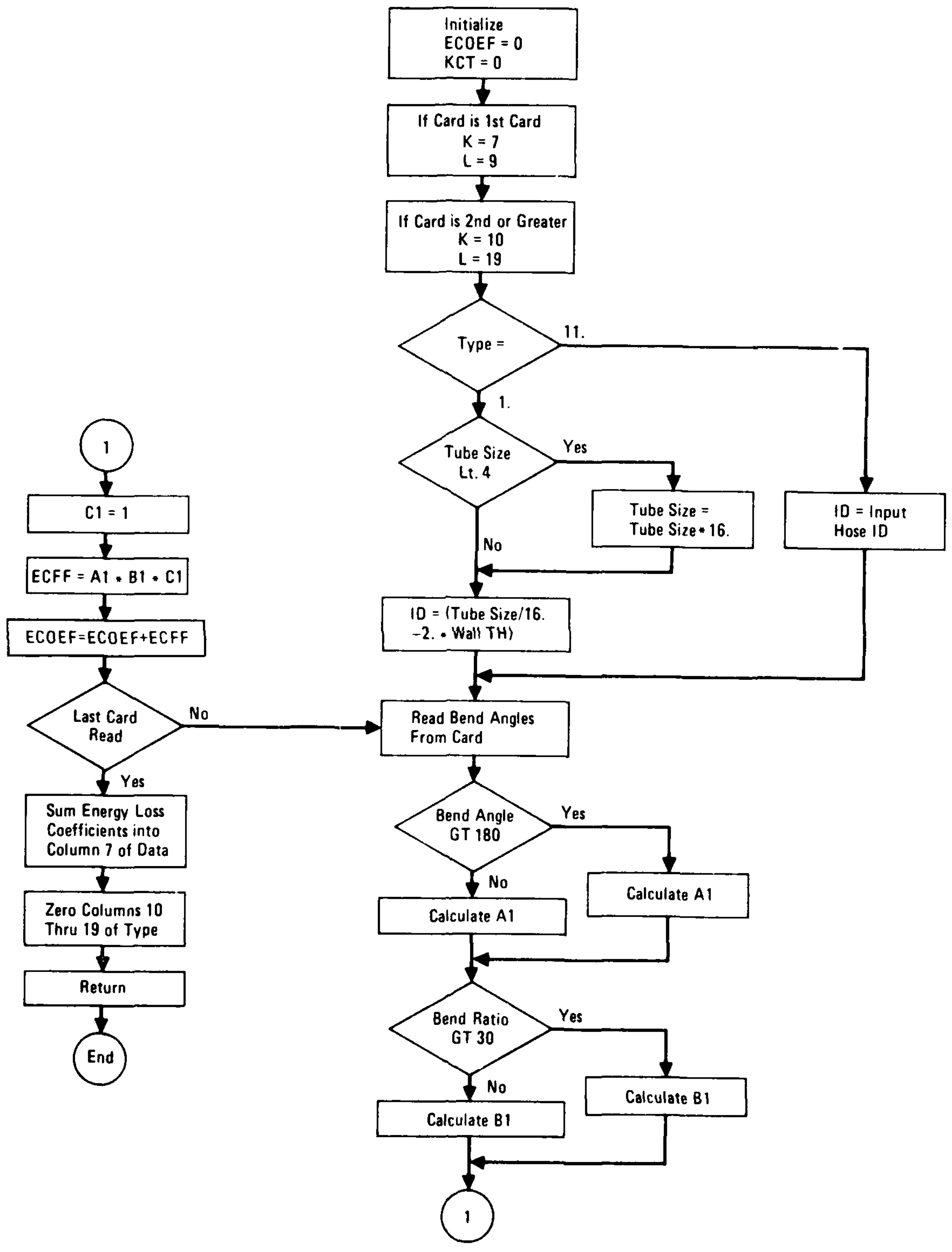
240.
245.
125.
130.
255.
270.
275.
280.
0.
0.
0.
0.
0.
0.
0.
0.
0.
0.
0.
0.

FIGURE 17

AFBP Array - Example Data

```
Row  1   { 4.000 145.000 135.000   4.000   4.000   4.500   3.250  35.0001000.000   5.000   2.500
         { 2.000  95.000  -0.000  -0.000  -0.000  -0.000   0.000   0.000   0.000   0.000   0.000
Row  2   { 4.000 130.000 170.000   4.000   4.000   3.350   2.000  25.000 100.000   5.000   2.500
         { 2.000  95.000  -0.000  -0.000  -0.000  -0.000   0.000   0.000   0.000   0.000   0.000
Row  3   {34.000  85.000  90.000  95.000 100.000   6.000   8.000   4.000   4.000  25.000 200.000
         {14.000    .1003000.000   1.000  -0.000  -0.000   0.000   0.000   0.000   0.000   0.000
Row  4   { 9.000 250.000 260.000 265.000  10.000   4.000  16.000   1.635  95.300   2.000  -0.000
         {-0.000  -0.000  -0.000  -0.000  -0.000  -0.000   0.000   0.000   0.000   0.000   0.000
Row  5  {    5.000   5.000  10.000  15.000  16.000  12.000   4.0003750.0003750.000  50.0003000.000
        {2950.000  -0.000  -0.000  -0.000  -0.000  -0.000   0.000   0.000   0.000   0.000   0.000
         {24.000 175.000 150.000 155.000   4.000   4.000   4.000  -0.000  -0.000   0.000   0.000
Row  6   { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
Row  7   {24.000 115.000 120.000 135.000   4.000   4.000   4.000  -0.000  -0.000   0.000   0.000
         { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
Row  8   {24.000  55.000  65.000  60.000   3.000   4.000   6.000  -0.000  -0.000   0.000   0.000
         { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
Row  9   {24.000 210.000 215.000 230.000   3.000   4.000  10.000  -0.000  -0.000   0.000   0.000
         { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
Row 10   { 3.000  50.000  55.000   8.000   8.000   5.000  -0.000  -0.000  -0.000   0.000   0.000
         { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
Row 11   { 6.000 240.000 245.000  10.000  10.000  15.000  10.000  18.000  37.000    .500 100.000
         {10.000  -0.000  -0.000  -0.000  -0.000  -0.000   0.000   0.000   0.000   0.000   0.000
Row 12   { 3.000  90.000 190.000   8.000   8.000   5.000  -0.000  -0.000  -0.000   0.000   0.000
         { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
Row 13   {24.000  35.000  45.000  40.000  12.000   4.000   8.000  -0.000  -0.000   0.000   0.000
         { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
Row 14   { 7.000 295.000   4.000   1.000  50.000 100.000   4.000   6.300   0.000   0.000   0.000
         { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
Row 15   { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
         { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
Row 16   { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
         { 0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
   (etc.)  0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
```

FIGURE 18 BRANCHP Array - Example Data



GP03-0415-9

FIGURE 19
BLOSS FLOW DIAGRAM

4.1.3 Error Messages

Should the user insert an invalid element number, the type number (ITYPE), generated from the integer truncation of TYPE(1) will cause an error message to be printed.

The message "Error - Illegal Element Type" is printed below the data, see Figure 20.

```
1          1.000 225.000 220.000   4.000    .020  78.200  97.000  89.000  90.000
1          2.000  15.000 225.000   4.000   4.000  -0.000  -0.000  -0.000  -0.000
1         26.000 200.000 205.000   8.000   8.000  -0.000  -0.000  -0.000  -0.000
           ***ERROR*** ILLEGAL ELEMENT TYPE
1          1.000 190.000 200.000   8.000    .028 128.500  97.000  31.000  68.000
```

FIGURE 20
Illegal Element Number Output

If data is entered in columns 2 thru 8 of any data card, an error message will be generated. The error message will be printed immediately below the input data, as shown in Figure 21.

```
2          1.000 205.000 210.000   8.000    .028 147.300  23.000  38.000  90.000
  83.000 121.000 150.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000
2    5     5.000   5.000  10.000  15.000  16.000  12.000   4.0003750.0003750.000
     ***ERROR*** DATA IN COLUMNS 2 THRU 8
  50.0003000.0002950.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000  -0.000
1          2.000 245.000 250.000  10.000  10.000  -0.000  -0.000  -0.000  -0.000
```

FIGURE 21

Illegal Format Output

The number of elements should not exceed the array storage allocated for them in the main program. If this occurs the excess elements of a particular type will be stored in the last location of the element array over any information that has previously been placed there. With termination of the data input operation system assembly begins.

4.1.4 Main Program Data Input Section Variable Names

Refer to section 3.7 for a variable name listing.

4.1.5 Main Program Data Input Section Listing

```
      READ(5,7)(FTYPE(M),M=1,8)
      READ(5,7)(FTYPE(M),M=9,16)
      CALL OPUT4
      WRITE(6,2)
      WRITE(6,1)
    1 FORMAT(1X,9('0'),10('1'),10('2'),10('3'),10('4'),10('5'),10('6'),
     110('7'),'8')
    2 FORMAT(' ',33('*'),'COLUMN NUMBERS',33('*'))
      WRITE(6,3)
    3 FORMAT(' ',8('1234567890'))
      WRITE(6,4)
    4 FORMAT(' ',5(8('+'),8('$')))
      ERR=0.
      NJ3=0
      N16=0
      N17=0
      N18=0
      N19=0
      READ(5,12)NVIS,BLANK1(1),(VVISC(M),M=1,NVIS)
      READ(5,12)NVIS,BLANK2(1),(VTEMP(M),M=1,NVIS)
      READ(5,15)DDENS(1),DDENS(2),DTEMP(1),DTEMP(2)
      READ(5,15)TEMP,TEMPF,TEMPINC,ALT,(PRINT(M),M=1,4)
      WRITE(6,5)
    5 FORMAT(/' CARD 1 ',23('+'),'TITLE',45('+'))
      WRITE(6,8)(FTYPE(M),M=1,8)
      WRITE(6,6)
    6 FORMAT(/' CARD 2 ',23('+'),'FLUID',45('+'))
      WRITE(6,8)(FTYPE(M),M=9,16)
    7 FORMAT(8A10)
      WRITE(6,9)
    8 FORMAT(' ',8A10)
    9 FORMAT(/' CARDS 3&4 ',9('+'),'VISCOSITY-TEMPERATURE DATA',35('+'))
      WRITE(6,13)NVIS,BLANK1(1),(VVISC(M),M=1,NVIS)
      WRITE(6,13)NVIS,BLANK2(1),(VTEMP(M),M=1,NVIS)
      WRITE(6,10)
   10 FORMAT(/' CARD 5 ',14('+'),'DENSITY-TEMPERATURE DATA',35('+'))
      WRITE(6,11)DDENS(1),DDENS(2),DTEMP(1),DTEMP(2)
   11 FORMAT(' ',10F8.3)
   12 FORMAT(I1,A7,9F8.2)
   13 FORMAT(' ',I1,A7,9F8.2)
      WRITE(6,14)
   14 FORMAT(/' CARD 6 ',5('+'),'INITIAL TEMP-FINAL TEMP-TEMP INCR-ALTIT
     1UDE-PRINT-OPTIONS',12('+'))
      WRITE(6,11)TEMP,TEMPF,TEMPINC,ALT,(PRINT(M),M=1,4)
   15 FORMAT(10F8.3)
      WRITE(6,16)
   16 FORMAT(//11X,'TYPE',2X,'DESCRIPTION',4X,'TYPE',
     12X,'DESCRIPTION',4X,'TYPE',2X,'DESCRIPTION',
     2/11X,4('+'),2X,11('+'),4X,4('+'),2X,11('+'),
     34X,4('+'),2X,11('+'))
```

4.1.5 (Continued)

```
      DO 17 I=1,10
   17 WRITE(6,18)ITY(I),IDES(I),ITY(I+10),IDES(I+10),ITY(I+20),
     1IDES(I+20)
   18 FORMAT(12X,I2,3X,A10,6X,I2,3X,A10,6X,I2,3X,A10)
      WRITE(6,19)
   19 FORMAT('1CARDS 7 & ON ',27('+'),'ELEMENT DATA',28('+'))
      WRITE(6,2)
      WRITE(6,1)
      WRITE(6,3)
      WRITE(6,4)
   20 READ(5,21)N,BLANK3(1),(TYPE(M),M=1,9)
      NCT=1
   21 FORMAT(I1,A7,9F8.3)
      WRITE(6,22)N,BLANK3(1),(TYPE(M),M=1,9)
   22 FORMAT(' ',I1,A7,9F8.3)
      AN=TYPE(1)
      IF(AN.EQ.1..OR.AN.EQ.11.)CALL BLOSS(NCT,TYPE)
      IF(N.GE.2)READ(5,15)(TYPE(M),M=10,19)
      IF(N.GE.2)NCT=2
      IF(BLANK3(1).EQ.BCHK)GO TO 24
      ERR=ERR+1.
      WRITE(6,23)
   23 FORMAT(' ',5X,'^**ERROR*** DATA IN COLUMNS 2 THRU 8')
   24 ITYPE=TYPE(1)
      IF(ITYPE.EQ.0)GO TO 58
      IF(ITYPE.GT.10)GO TO 25
      IF(ITYPE.GT.2)GO TO 35
      GO TO (37,37),ITYPE
   25 J1=ITYPE/10
      J2=ITYPE-J1*10
      GO TO (26,27,28,30,30,30,30,30,29),J1
   26 GO TO (37,37,52,32,41,41,41,41,41),J2
      GO TO 30
   27 GO TO (37,37,37,35,35),J2
      GO TO 30
   28 GO TO (35,37,35,35,35,35,35,35),J2
      GO TO 30
   29 GO TO (35,35),J2
   30 ERR=ERR+1.
      WRITE(6,31)
   31 FORMAT(' ',11X,'^**ERROR*** ILLEGAL ELEMENT TYPE')
      GO TO 56
   32 DO 33 J3=3,9
      IF(TYPE(J3).EQ.0.)GO TO 34
      NJ3=NJ3+1
      TEMP14(NJ3,2)=TYPE(2)
   33 TEMP14((NJ3),1)=TYPE(J3)
   34 GO TO 54
   35 NBP=NBP+1
      IF(N.EQ.2)WRITE(6,57)(TYPE(M),M=10,19)
      DO 36 J=1,17
   36 BRANCHP(NBP+(J-1)*NBP2)=TYPE(J)
      GO TO 54
   37 NC=NC+1
      IF(N.GE.2)WRITE(6,57)(TYPE(M),M=10,19)
```

4.1.5 (Continued)

```
      IF(N.GE.2)CALL BLOSS(NCT,TYPE)
      IF(N.EQ.NCT)GO TO 39
   38 READ(5,15)(TYPE(M),M=10,19)
      WRITE(6,57)(TYPE(M),M=10,19)
      CALL BLOSS(NCT,TYPE)
      NCT=NCT+1
      IF(NCT.EQ.N)GO TO 39
      GO TO 38
   39 DO 40 J=1,8
   40 CONNECT(NC+(J-1)*NC2)=TYPE(J)
      GO TO 54
   41 IF(N.EQ.2)WRITE(6,57)(TYPE(M),M=10,19)
      J3=ITYPE-14
      GO TO (42,44,46,48,50),J3
   42 DO 43 I=1,3
      J=I+1
   43 QT15(I)=TYPE(J)
      GO TO 54
   44 N16=N16+1
      DO 45 I=1,18
      J=I+1
   45 QT16(N16,I)=TYPE(J)
      GO TO 54
   46 N17=N17+1
      DO 47 I=1,18
      J=I+1
   47 QT17(N17,I)=TYPE(J)
      GO TO 54
   48 N18=N18+1
      J=I+1
   49 QT18(N18,I)=TYPE(J)
      GO TO 54
   50 N19=N19+1
      DO 51 I=1,5
      J=I+1
   51 QT19(I,N19)=TYPE(J)
      GO TO 54
   52 DO 53 J=2,9
      IF(TYPE(J).EQ.0.)GO TO 54
      NAB=NAB+1
   53 AFBP(NAB)=TYPE(J)
   54 DO 55 M=1,19
   55 TYPE(M)=0.
      GO TO 20
   56 IF(N.EQ.2)WRITE(6,57)(TYPE(M),M=10,19)
      GO TO 20
   57 FORMAT(' ',10F8.3)
   58 CONTINUE
      CALL VISD
      IF(ALT.LE.36089.)PAMB=(1.6676*(1.-ALT/145378.))**5.256
      IF(ALT.GT.36089.AND.ALT.LE.65E3)PAMB=3.2825
     1 *EXP(4.806E-5*(36089.-ALT))
      IF(ALT.GT.65E3.AND.ALT.LE.15E4)PAMB=(867.42375-1.990498E-2*ALT
     1 +1.549619E-7*(ALT**2)-4.046556E-13*(ALT**3))/144.
      IF(ALT.GT.15E4)PAMB=.0197361
```

4.1.6 BLOSS Subroutine Variable Names

Variable	Description	Units
A1	Part of energy loss coefficient	--
B	Bend angle	DEG.
B1	Part of energy loss coefficient	--
BRAT	Bend Ratio	--
C1	Part of energy loss coefficient	--
DIA	Tube of hose I.D.	IN
ECFF	Energy loss coefficient for 1 bend	--
ECOEF	Summation of individual energy loss coefficients ECFF	--
K	Location for first bend angle to be read in TYPE array	--
KCT	Counter to determine whether or not the card processed is the first card	--
L	Location for last bend angle to be read in TYPE array	--
M	Counter	--
NCT	Indicator from main program to determine which data to process	--
TYPE	Temporary data storage array	--

4.1.7 BLOSS Subroutine Listing

```
  SUBROUTINE BLOSS(NCT,TYPE)
  DIMENSION TYPE(1)
  ECOEF=0.
  KCT=0
  L=9
  K=7
  IF(NCT.GT.1)K=10
  IF(NCT.GT.1)L=19
  IF(TYPE(1).EQ.11.)GO TO 1
  IF(TYPE(4).LT.4.)TYPE(4)=TYPE(4)*16.
  DIA=TYPE(4)/16.-2.*TYPE(5)
  BRAT=3.*(TYPE(4)/16.)/DIA
  GO TO 2
1 DIA=TYPE(5)
  BRAT=8.0
2 DO 7 I=K,L
  KCT=KCT+1
  B=TYPE(I)
  IF(NCT.EQ.1.AND.KCT.EQ.1)TYPE(7)=0.
  IF(B.EQ.0.)GO TO 7
  IF(B.GT.180.)GO TO 3
  A1=-3.97626E-10*B**4+2.8475E-7*B**3-9.23298E-5*B**2
 1+1.74517E-2*B
  GO TO 4
3 A1=1.39+(B-180.)*3.3333E-3
4 IF(BRAT.GT.30.)GO TO 5
  B1=.206982*BRAT**(-.49421)
  GO TO 6
5 B1=.0335411-(BRAT-30.)*4E-4
6 C1=1.
  ECFF=A1*B1*C1
  ECOEF=ECOEF+ECFF
7 CONTINUE
  TYPE(7)=TYPE(7)+ECOEF
  DO 8 M=10,19
  IF(NCT.GT.1)TYPE(M)=0.
8 CONTINUE
  RETURN
  END
```

SECTION V

SYSTEM ASSEMBLY AND INITIAL CALCULATION

The system is assembled into legs from the individual elements. Starting with a pump or Type 7 accumulator flow or pressure source, a leg is assembled using a continuity search for the element containing the matching junction number. As each element is located, static resistance factors are calculated and stored. When a match is found and the element is a branch point element, the leg is ended and a new leg is started. If an element is a dynamic element (elements whose data are updated during the iteration calculation) identifiers are placed in the element data array. This procedure continues until all legs are assembled.

Figure 22 is a general flow diagram for system assembly.

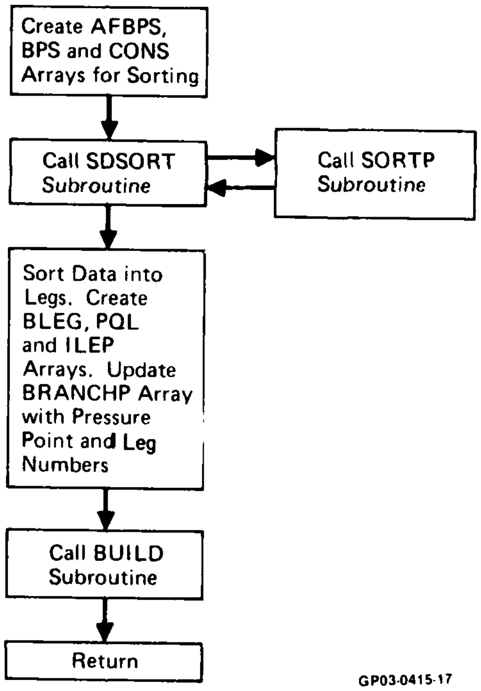

Create AFBPS, BPS and CONS Arrays for Sorting
Call SDSORT Subroutine
Call SORTP Subroutine
Sort Data into Legs. Create BLEG, PQL and ILEP Arrays. Update BRANCHP Array with Pressure Point and Leg Numbers
Call BUILD Subroutine
Return
GP03-0415-17

FIGURE 22
SYSTEM ASSEMBLY FLOW DIAGRAM

5.1 SYSTEM ASSEMBLY SUBROUTINES

Subroutine SDSORT takes the element type number and junction number data from arrays AFBP, BRANCHP and CONNECT, Figures 16, 17 and 18 and places the data in arrays AFBPS, BPS and CONS. The AFBPS, BPS and CONS arrays are used during the leg building process only, and the data contained therein is destroyed as the legs are built, see Figures 23 and 24 for before and after leg building. The data in BPS array is arranged according to ascending element type number with types 5 and 7 placed at the beginning of the array. Column 6 of BPS array contains the row location of the element in BRANCHP array. SORTP subroutine is used to sort the data in BPS array in ascending order by type number except that type 5 and type 7 data are placed at the top of the array. This minimizes the search for connecting elements during the building process.

Subroutine BUILD performs the leg building. The first element in BPS array is used to start the first leg if it is an element type 5 or type 7. If no element type 5 or type 7 is available, an error message is printed, see Figure 25. A search is made for the element having a matching junction number. When a matching element junction is found, the outlet port junction of the new element is set as the port junction search number. If a branch point element junction is in the junction match, the leg is ended and a new leg is started. A new leg will not be started from a branch point element unless there has already been a junction assembled from that element or the element is a type 5 or type 7. Therefore inactive elements may be left in the data cards when checking different system configurations, and will not be assembled into the system as long as the junction numbers do not match any of the active system element numbers.

```
  1.000  40.000  50.000            1.000   0.000   0.000
  1.000 230.000 235.000            1.000   0.000   0.000
  1.000 205.000 210.000            1.000   0.000   0.000
  2.000 245.000 250.000            2.000   0.000   0.000
  2.000 235.000 240.000            2.000   0.000   0.000
  2.000 220.000 215.000            2.000   0.000   0.000
  1.000 225.000 220.000            1.000   0.000   0.000
  2.000  15.000 225.000            2.000   0.000   0.000
  2.000 200.000 205.000            2.000   0.000   0.000
  1.000 190.000 200.000            1.000   0.000   0.000
  1.000 165.000 160.000   CONS     1.000   0.000   0.000
  2.000 170.000 165.000  Arrays    2.000   0.000   0.000
  2.000 150.000 100.000            2.000   0.000   0.000
  1.000 155.000 150.000            1.000   0.000   0.000
  1.000 180.000 175.000            1.000   0.000   0.000
  2.000 185.000 180.000            2.000   0.000   0.000
  2.000 140.000 145.000            2.000   0.000   0.000
  1.000 135.000 140.000            1.000   0.000   0.000
  2.000 125.000 130.000            2.000   0.000   0.000
  1.000 120.000 125.000            1.000   0.000   0.000
  1.000 110.000 115.000            1.000   0.000   0.000
  2.000  95.000 110.000            2.000   0.000   0.000
  2.000  70.000  75.000            2.000   0.000   0.000
  1.000  75.000  80.000            1.000   0.000   0.000
  1.000  60.000  70.000            1.000   0.000   0.000
  2.000 290.000 295.000            2.000   0.000   0.000
  1.000  65.000 290.000            1.000   0.000   0.000
  1.000  30.000  35.000            1.000   0.000   0.000
 11.000  20.000  25.000           11.000   0.000   0.000
 21.000  80.000  85.000           21.000   0.000   0.000
  1.000  45.000 255.000            1.000   0.000   0.000
  2.000 255.000 260.000            2.000   0.000   0.000
 11.000 280.000 285.000           11.000   0.000   0.000
 22.000  10.000  20.000           22.000   0.000   0.000
 22.000 265.000 270.000           22.000   0.000   0.000
  1.000 270.000 275.000            1.000   0.000   0.000
  2.000 275.000 280.000            2.000   0.000   0.000
 21.000  25.000  30.000           21.000   0.000   0.000
  2.000 285.000   5.000            2.000   0.000   0.000
  0.000   0.000   0.000            0.000   0.000   0.000

        before                             after
```

```
240.              -1.
245.              -1.
125.    AFBPS     -1.
130.    Arrays    -1.
255.              -1.
270.              -1.
275.              -1.
280.              -1.
  0.               0.
  0.               0.
  0.               0.
  0.               0.
  0.               0.
  0.               0.
  0.               0.
  0.               0.
  0.               0.
  0.               0.
  0.               0.
  0.               0.

before           after
```

FIGURE 23 CONS and AFBPS Arrays- Before and After Leg Assembly

```
 5.000   5.000  10.000  15.000    0.000   5.000
 7.000 295.000   4.000   1.000   50.000  14.000
 3.000  50.000  55.000   0.000    0.000  10.000
 3.000  90.000 190.000   0.000    0.000  12.000
 4.000 145.000 185.000   0.000    0.000   1.000
 4.000 130.000 170.000   0.000    0.000   2.000
 6.000 240.000 245.000   0.000    0.000  11.000      before
 9.000 250.000 260.000 265.000    0.000   4.000       leg
24.000 210.000 215.000 230.000    0.000   9.000      assembly
24.000 115.000 120.000 135.000    0.000   7.000
24.000  55.000  65.000  60.000    0.000   8.000
24.000  35.000  45.000  40.000    0.000  13.000
24.000 175.000  60.000 155.000    0.000   6.000
34.000  85.000  90.000  95.000  100.000   3.000


 5.000  -1.000  -1.000  -1.000    0.000   5.000
 7.000  -1.000   0.000   0.000    0.000  14.000
 3.000  -2.000  -2.000   0.000    0.000  10.000
 3.000  -2.000  -2.000   0.000    0.000  12.000
 4.000  -1.000  -1.000   0.000    0.000   1.000
 4.000  -1.000  -1.000   0.000    0.000   2.000      after
 6.000  -2.000  -2.000   0.000    0.000  11.000       leg
 9.000  -1.000  -1.000  -1.000    0.000   4.000      assembly
24.000  -1.000  -1.000  -1.000    0.000   9.000
24.000  -1.000  -1.000  -1.000    0.000   7.000
24.000  -1.000  -1.000  -1.000    0.000   8.000
24.000  -1.000  -1.000  -1.000    0.000  13.000
24.000  -1.000  -1.000  -1.000    0.000   6.000
34.000  -1.000  -1.000  -1.000   -1.000   3.000
```

FIGURE 24 BPS Array

As each element is encountered in building a leg, static resistance coefficients are calculated by SCONST1, SCONST2 or SCONST3 subroutine. These resistance coefficients are summed into BLEG array, columns 8, 9, 10 and 11, see Figure 7 . SCONST1 calculates static resistance coefficients for connecting type elements all stored in CONNECT array. SCONST2 calculates static resistance coefficients for elements in BRANCHP array which are not branch point elements such as check valves, filters, etc. SCONST3 calculates static resistance coefficients for all branch point elements stored in BRANCHP array.

During assembly, branch point numbers are placed in the BRANCHP array for all elements that are branch point elements, see Figure 26 . A (-1) is placed in column 21 for these same elements. For elements that are not branch point elements (Types 3, 6, 10, 31 and 33) in BRANCHP array, the leg number in which it is assembled is placed in column 21. The column 21 identifiers are used to indicate leg placement of calculations. If a (-1.) or leg number is not placed in column 21, the element is inactive and will not be called for calculation during the iteration calculation program section.

```
***ERROR*** NO PUMP OR TYPE 7 PRESSURE OR FLOW SOURCE
```

FIGURE 25 Error Message - No element type 5 or type 7 available

```
         [illegible] 155.000     5.000     6.000     4.500     3.250   35.0001000.000     5.000     2.500
         [illegible]  -0.000    -0.000    17.000    -0.000     0.000     0.000     0.000    -1.000     0.000
[illegible] 130.000  170.000     7.000     8.000     3.850     2.900   25.0001000.000     5.000     2.500
[illegible]  95.000   -0.000    -0.000    18.000    -0.000     0.000     0.000     0.000    -1.000     0.000
[illegible]  55.000   90.000    95.000   100.000    17.000    13.000    19.000    20.000    25.000   200.000
[illegible]    .1003000.000     1.000    19.000    20.000    21.000    22.000    23.000    -1.000     0.000
[illegible] 250.000  260.000   265.000     9.000    10.000    11.000     1.655    95.300     2.000     4.000
    5.000     2.000   -1.000    -0.000    -0.000     0.000     0.000     0.000     0.000    -1.000     0.000
    5.000     5.000   10.000    15.000     1.000     2.000     3.0003750.0003750.000    50.0003000.000
[illegible]  -0.000   -0.000    -0.000    -0.000     1.000    12.351    50.000     0.000    -1.000     0.000
   24.000   175.000  150.000   155.000      .172      .172      .172    15.000    13.000    16.000     0.000
    0.000     1.000    0.000     0.000     0.000     0.000     0.000     0.000     0.000    -1.000     0.000
   24.000   115.000  120.000   135.000      .172      .172      .172     1.000    12.000    14.000     0.000
    1.000     1.000    0.000     0.000     0.000     0.000     0.000     0.000     0.000    -1.000     0.000
   24.000    55.000   65.000    60.000      .391      .172      .297     8.000     9.000    10.000     0.000
    1.000     1.000    0.000     0.000     0.000     1.000     0.000     0.000     0.000    -1.000     0.000
   24.000   210.000  215.000   230.000      .391      .172      .484     7.000     3.000     5.000     1.000
    0.000     0.000    0.000     0.000     0.000     0.00      0.000     0.000     0.000    -1.000     0.000
    3.000    50.000   55.000      .391      .391     5.000     1.000    -0.000    -0.000     0.000     0.000
    0.000     0.000    0.000     0.000     0.000     0.000     0.000     0.000     0.000     8.000     0.000
    5.000   240.000  245.000      .484      .484    15.000     2.000    18.000    37.000      .500   100.000
   10.000     2.000   -0.000    -0.000    -0.000    -0.000     0.000     0.000     0.000     4.000     0.000
    3.000    90.000  100.000      .391      .391     5.000     2.000    -0.000    -0.000     0.000     0.000
    0.000     0.000    0.000     0.000     0.000     0.000     0.000     0.000     0.000     7.000     0.000
   24.000    35.000   45.000    40.000      .609      .172      .391     1.000     6.000     8.000     0.000
    0.000     1.000    0.000     0.000     0.000     0.000     0.000     0.000     0.000    -1.000     0.000
    7.000   295.000    4.000     1.000    50.000   100.000     4.000     6.800     0.000     0.000     0.000
    0.000     0.000    0.000     0.000     0.000     0.000     0.000     0.000     0.000    -1.000     0.000
    0.000     0.000    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
    0.000     0.000    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
```

FIGURE 26 BRANCHP Array - After Leg Assembly

5.1.1 <u>SDSORT AND SORTP SUBROUTINES</u>

SDSORT subroutines builds the AFBPS, BPS and CONS arrays which are used for leg building. The junction numbers from AFBP are placed directly into AFBPS. Element type and junction numbers are taken from BRANCHP and CONNECT and placed in BPS and CONS respectively. Additionally, the data in BPS is placed in ascending order by element type through SORTP subroutine except that types 5 and 7 are placed at the beginning of the array. Column 6 of BPS is used to cross-reference the line location of the element in BRANCHP array. Figure 27 presents a generalized flow diagram of SDSORT.

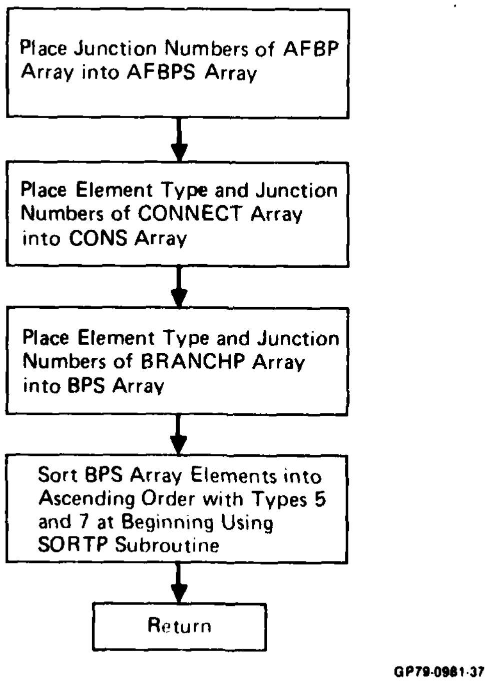

Place Junction Numbers of AFBP Array into AFBPS Array
Place Element Type and Junction Numbers of CONNECT Array into CONS Array
Place Element Type and Junction Numbers of BRANCHP Array into BPS Array
Sort BPS Array Elements into Ascending Order with Types 5 and 7 at Beginning Using SORTP Subroutine
Return
GP79-0981-37

FIGURE 27 SDSORT-SORTP FLOW DIAGRAM

5.1.1.1 SDSORT AND SORTP VARIABLE NAMES

Variables	Description	Dimensions
AFBP	Array Containing Type 13 Junction Numbers	--
AFBPS	Array Containing Type 13 Junction Numbers	--
BPS	Array Containing Dynamic Element Types and Junction Numbers	--
BRANCHP	Array Containing all Data for Dynamic Elements	--
CONNECT	Array Containing all Data for Static Elements	--
CONS	Array Containing Static Element Types and Junction Numbers	--
DBP	Variable Used for Temporary Storage of BPS Array Variable During Sorting Element Types	--
I	Do Loop Counter	--
ITYPE	Element Type	--
J	Do Loop Counter	--
J1	Integer Value of Element Type Divided by 10	--
J2	Unit's Integer Value of Element Type	--
MAXT	Maximum Value of Element Type Used in System Data	--
N	Indicator Used for Each Dynamic Element for Transfer of Junction Numbers to BPS Array from BRANCHP Array	--
NBP	Counter Used for Current Location in BPS Array	--
NBP1	Counter Used to Search all Locations beyond NBP Location in BPS Array	--
NBP2	Total Length of BRANCHP Array	--
NC2	Total Length of Connect Array	--
NPA	Count of Number of Type 5 and Type 7 Elements	--
NPP	Actual Count of Number of Elements in BPS	--

5.1.1.2 <u>SDSORT and SORTP Subroutine Listings</u>

```
      SUBROUTINE SDSORT(AFBP,BRANCHP,CONNECT,AFBPS,BPS,CONS,NBP2,NC2)
C        BUILD ARRAYS FOR SORTING DATA                   5/23/79
      DIMENSION AFBP(1),BRANCHP(1),CONNECT(1)
      DIMENSION AFBPS(1),BPS(1),CONS(1)
      DO 1 I=1,1000
      IF(AFBP(I).EQ.0.)GO TO 2
    1 AFBPS(I)=AFBP(I)
    2 DO 3 I=1,10000
      IF(CONNECT(I).EQ.0.)GO TO 4
      DO 3 J=1,3
    3 CONS(I+(J-1)*NC2)=CONNECT(I+(J-1)*NC2)
    4 DO 16 I=1,10000
      IF(BRANCHP(I).EQ.0.)GO TO 17
      ITYPE=BRANCHP(I)
      IF(ITYPE.GT.10)GO TO 5
      GO TO (7,7,11,11,12,11,10,12,12,11),ITYPE
    5 IF(ITYPE.EQ.11)GO TO 7
      IF(ITYPE.EQ.12)GO TO 11
      J1=ITYPE/10
      J2=ITYPE-J1*10
      IF(J1.GT.3)GO TO 11
      GO TO (7,6,9),J1
    6 IF(ITYPE.EQ.25)GO TO 13
      GO TO 12
    7 WRITE(6,8)ITYPE
    8 FORMAT('ILLEGAL ELEMENT TYPE',I4)
      GO TO 16
    9 IF(ITYPE.EQ.34)GO TO 13
      IF(ITYPE.EQ.35)GO TO 12
      GO TO 11
   10 N=5
      GO TO 14
   11 N=3
      GO TO 14
   12 N=4
      GO TO 14
   13 N=5
   14 DO 15 J=1,N
   15 BPS(I+(J-1)*NBP2)=BRANCHP(I+(J-1)*NBP2)
      BPS(I+5*NBP2)=I
   16 CONTINUE
   17 CONTINUE
      NPA=0
      DO 18 I=1,10000
      IF(BPS(I).EQ.0.)GO TO 19
      IF(BPS(I).EQ.7..OR.BPS(I).EQ.5.)NPA=NPA+1
   18 CONTINUE
   19 IF(NPA.EQ.0)GO TO 27
      NBP=0
      DO 20 I=1,10000
```

5.1.1.2 (Continued)

```
      IF(BPS(I).EQ.0.)GO TO 21
      IF(BPS(I).NE.5.)GO TO 20
      CALL SORTP(I,BPS,NBP,NBP2)
   20 CONTINUE
   21 DO 22 I=1,10000
      IF(BPS(I).EQ.0.)GO TO 23
      IF(BPS(I).NE.7.)GO TO 22
      CALL SORTP(I,BPS,NBP,NBP2)
   22 CONTINUE
   23 MAXT=BPS(1)
      NPP=0
      DO 24 I=1,10000
      IF(BPS(I).EQ.0.)GO TO 25
      NPP=NPP+1
      IF(BPS(I).GT.MAXT)MAXT=BPS(I)
   24 CONTINUE
   25 DO 26 J=3,MAXT
      NBP1=NBP+1
      DO 26 I=NBP1,NPP
      IF(BPS(I).NE.J)GO TO 26
      CALL SORTP(I,BPS,NBP,NBP2)
   26 CONTINUE
      GO TO 29
   27 WRITE(6,28)
   28 FORMAT('***ERROR*** NO PUMP OR TYPE 7 PRESSURE OR FLOW SOURCE')
      RETURN
   29 CONTINUE
      RETURN
      END
      SUBROUTINE SORTP(I,BPS,NBP,NBP2)
      DIMENSION BPS(1)
      NBP=NBP+1
      DO 1 J=1,6
      DBP=BPS(NBP+(J-1)*NBP2)
      BPS(NBP+(J-1)*NBP2)=BPS(I+(J-1)*NBP2)
    1 BPS(I+(J-1)*NBP2)=DBP
      RETURN
      END
```

5.1.2 BUILD SUBROUTINE

Subroutine BUILD assembles the elements into legs, see Figure 28 for flow diagram. An initial search is made of AFBPS and BPS to determine whether any junction point number input into AFBPS is contained in BPS. If there is a duplication, the junction number in AFBPS is set to (-1.). PQL array is built next. Each branch point element in BPS array is assigned pressure point number(s) in PQL by its row number. If the pressure point is used as a fixed pressure for the iteration a positive pressure valve is assigned. If the pressure is calculated during the iteration, a (-1.) is assigned, see Figure 6. AFBPS is then searched and all remaining junction numbers (ones that have not been set to -1.) are assigned pressure point numbers.

The first leg is started with a type 5 pump or type 7 flow, pressure or accumulator source. To begin a leg, BEGM is set to 1. This is used as an indicator for some elements. SCONST1, SCONST2 and SCONST3 are called for the appropriate elements to place resistance factors in BLEG array. At the end of a leg BEGM is set to 0. This also is used as an indicator. Leg building continues until no more ports are available. After all element legs are assembled, internal legs are added for type 4, 8, 36, 37 and 38.

ILEP array is next built. This array contains the pressure point number at the end of each leg. The row location in ILEP corresponds to the row in BLEG. Figure 29 is an example of the leg assembly output data.

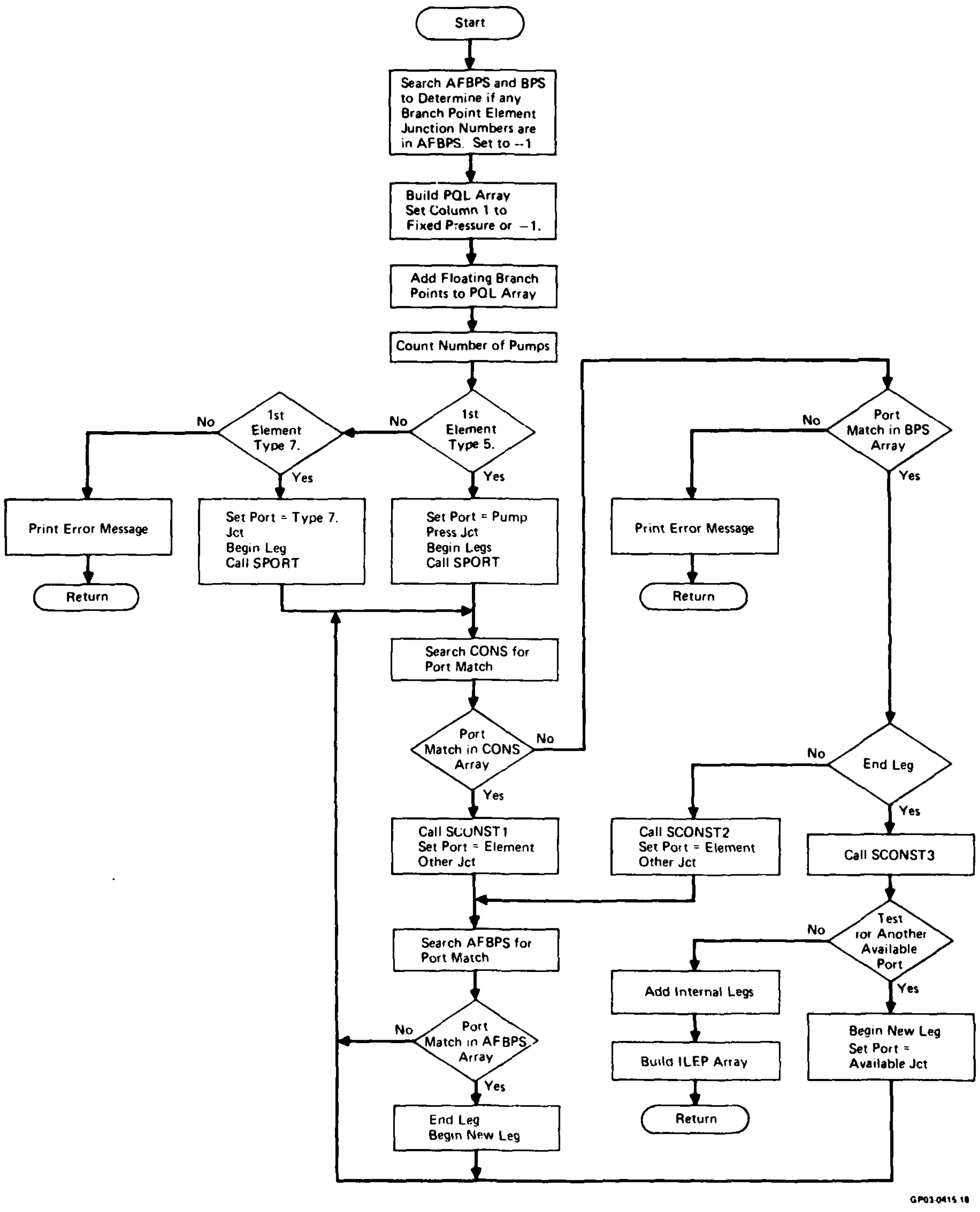

Start
Search AFBPS and BPS to Determine if any Branch Point Element Junction Numbers are in AFBPS. Set to –1
Build PQL Array Set Column 1 to Fixed Pressure or –1.
Add Floating Branch Points to PQL Array
Count Number of Pumps
1st Element Type 5.
No
Yes
1st Element Type 7.
No
Yes
Print Error Message
Return
Set Port = Type 7. Jct Begin Leg Call SPORT
Set Port = Pump Press Jct Begin Legs Call SPORT
Search CONS for Port Match
Port Match in CONS Array
No
Yes
Port Match in BPS Array
No
Yes
Print Error Message
Return
End Leg
No
Yes
Call SCONST1 Set Port = Element Other Jct
Call SCONST2 Set Port = Element Other Jct
Call SCONST3
Search AFBPS for Port Match
Port Match in AFBPS Array
No
Yes
End Leg Begin New Leg
Test for Another Available Port
No
Yes
Add Internal Legs
Build ILEP Array
Return
Begin New Leg Set Port = Available Jct
GP03-0415-18

FIGURE 28
SUBROUTINE BUILD FLOW DIAGRAM

```
LEG NO-LEG END PRESSURE PT NO-JUNCTION PT NO & ELEMENT TYPE

LEG  PRESSURE PT   BRANCH/END PT * PRESS  ELEMENT       JUNCTION
NO.  (UP)   (DN)   (UP)     (DN) * PT NO                 NUMBERS
 1     2     15     10.      35. *    1    PUMP             5.
 2     1     27      5.     280. *    2    PUMP            10.
 3    27     26    280.     275. *    3    PUMP            15.
 4    26     25    275.     270. *    4    ACCUM          295.
 5    25     11    270.     265. *    5    ACTR           145.
 6     3     12     15.     215. *    6    ACTR           185.
 7     9     22    250.     245. *    7    ACTR           130.
 8    22     21    245.     240. *    8    ACTR           170.
 9    21     12    240.     230. *    9    RESV           250.
10    10     24    260.     255. *   10    RESV           260.
11    24     15    255.      45. *   11    RESV           265.
12    12     18    210.      90. *   12    TEE      210.- 215.- 230.
13    15     14     40.      55. *   13    TEE      115.- 120.- 135.
14    14      4     65.     295. *   14    TEE       55.-  65.-  60.
15    14     17     60.      85. *   15    TEE       35.-  45.-  40.
16    19     13     95.     115. *   16    TEE      175.- 160.- 155.
17    13     23    120.     125. *   17    VLV             85.
18    23      7    125.     130. *   18    VLV             90.
19     8     16    170.     160. *   19    VLV             95.
20    13      5    135.     145. *   20    VLV            100.
21     6     16    185.     175. *   21    FBP            240.
22    16     20    155.     100. *   22    FBP            245.
23     5      6    145.     185. *   23    FBP            125.
24     7      8    130.     170. *   24    FBP            255.
25    17     18     85.      90. *   25    FBP            270.
26    17     19     85.      95. *   26    FBP            275.
27    19     18     95.      90. *   27    FBP            280.
28    17     20     85.     100. *
29    20     18    100.      90. *
```

FIGURE 29 Leg Number and Pressure Point Number Assembly

5.1.2.1 BUILD Variable Names

Variable	Description	Dimension
A	Inlet port size	IN or 16th IN
AFBPS	Array containing type 13 junction numbers	--
ALEN	Total length	IN
BEGM	Market indciator for leg 0 = begin leg 1 = end leg	--
BJCT	Port Number	--
BLEG	General purpose array	--
BPS	Dynamic element junction storage array for sorting	--
BRANCHP	Dynamic element data storage array	--
BRP1	Branch point number at beginning of leg	--
BRP2	Branch point number at end of leg	--
CONNECT	Static element data storage array	--
CONS	Static element junction storage array for sorting	--
DIAL	Sum of diameter x length	IN^2
DIAO	Previous element diameter	IN
I	Integer counter	--
IBLOC	Element row location in BPS array	--
IBRLOC	Row location of element in BRANCHP array	--
ILEP	Array of leg numbers with up and downstream pressure points	--
I1	Location of port in AFBPS array	--
I1C	Indicator to check outlet port to see if a type 13 had been input	--
J,K	Integer counters	--
KBP	Counter for number of branch points	--

5.1.2.1 (Continued)

Variable	Description	Dimension
KP	Integer Counter	--
L	Integer Counter	--
LEG	Row number in BLEG array	--
LL	Integer counter	--
M	Integer counter	--
ML	Total number of legs	--
N	Integer counter	--
NBP2	Total length of BRANCHP array	--
NC2	Total length of CONS array	--
NL	Total length of BLEG and ILEP arrays	--
NN	Indicator for storing assembly direction in BRANCHP array	--
NP	Counter for number of pumps	--
NPC	Port column location in CONS array	--
NPQ	Total rows in PQL array	--
NT	Column indicator for storing leg numbers in BRANCHP array	--
N1	Indicator for number of junctions in element	--
PM1	Junction number at beginning of leg	--
PM2	Junction number at end of leg	--
PORT	Junction number currently being investigated	--
PQL	Array containing calculated pressures, element types and junction numbers	--
SUM	Summation of diameter x length	IN^2
SUM1	Summation of lengths	IN

5.1.2.1 (Continued)

Variable	Description	Dimension
TI	Indicator for element type 7 subtype	--
TLEN	Inlet port length	IN
TLN	Summation of inlet and outlet port lengths	IN
TLNS	Total leg length	IN
TY	Element type	--
TYL	Row location of junction number in PQL array	--

5.1.2.2 BUILD Subroutine Listing

```
      SUBROUTINE BUILD(ILEP,CONS,BPS,AFBPS,BLEG,PQL,NBP2,NC2,NL,NPQ,
     1BRANCHP,CONNECT,ML)
      DIMENSION AFBPS(1),BPS(1),CONS(1),BLEG(1),PQL(1),ILEP(1)
      DIMENSION BRANCHP(1),CONNECT(1)
      NP=0
      LEG=0
      DO 4 L=1,100
      IF(AFBPS(L).EQ.0.)GO TO 5
      DO 2 M=1,10000
      IF(BPS(M).EQ.0.)GO TO 4
      TY=BPS(M)
      IF(TY.EQ.3..OR.TY.EQ.6..OR.TY.EQ.10..OR.TY.EQ.31.
     1.OR.TY.EQ.33.)GO TO 2
      DO 1 N=2,5
      IF(BPS(M+(N-1)*NBP2).EQ.AFBPS(L))GO TO 3
    1 CONTINUE
    2 CONTINUE
      GO TO 4
    3 AFBPS(L)=-1.
    4 CONTINUE
    5 CONTINUE
      KBP=0
      DO 14 I=1,10000
      IF(BPS(I).EQ.0.)GO TO 15
      TY=BPS(I)
      IF(TY.EQ.3..OR.TY.EQ.6..OR.TY.EQ.10..OR.TY.EQ.31.
     1.OR.TY.EQ.33.)GO TO 14
      IF(TY.EQ.7..OR.TY.EQ.24..OR.TY.EQ.25.)GO TO 8
      IF(TY.EQ.4..OR.TY.EQ.36..OR.TY.EQ.37..OR.TY.EQ.38..OR.TY.EQ.91..O
     1R.TY.EQ.92.)GO TO 12
      IF(TY.EQ.5..OR.TY.EQ.8..OR.TY.EQ.9..OR.TY.EQ.35.)GO TO 13
      N1=5
C     ****BUILD PQL ARRAY***
    6 DO 7 J=2,N1
      IF(TY.EQ.8..AND.BPS(I+(J-1)*NBP2).EQ.0.)GO TO 14
      KBP=KBP+1
      PQL(KBP)=-1.
      PQL(NPQ+KBP)=I
      IF(TY.EQ.5..AND.J.EQ.3)PQL(KBP)=3000.
      IF((TY.EQ.9..OR.TY.EQ.91..OR.TY.EQ.92.).AND.J.EQ.2)PQL(KBP)=50.
      IF(TY.EQ.92..AND.J.EQ.3.)PQL(KBP)=50.
      IF(TY.EQ.9..AND.J.EQ.4)PQL(KBP)=50.
      PQL(KBP+2*NPQ)=TY
    7 PQL(KBP+3*NPQ)=BPS(I+(J-1)*NBP2)
      GO TO 14
    8 KBP=KBP+1
      PQL(KBP)=-1.
      PQL(NPQ+KBP)=I
      DO 9 J=1,5
    9 PQL(KBP+(J+1)*NPQ)=BPS(I+(J-1)*NBP2)
```

5.1.2.2 (Continued)

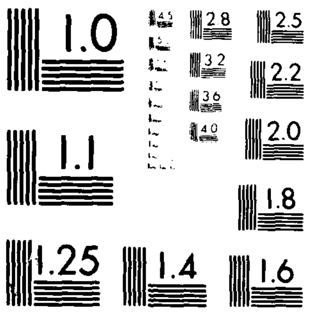
```
      IF(TY.EQ.24..OR.TY.EQ.25.)GO TO 14
      TI=BPS(I+3*NBP2)
      IF(TI.EQ.1..OR.TI.EQ.4.)GO TO 10
      PQL(KBP)=BPS(I+4*NBP2)
 10   DO 11 KP=2,4
      PQL(KBP+(2+KP)*NPQ)=0.
 11   BPS(I+KP*NBP2)=0.
      GO TO 14
 12   N1=3
      GO TO 6
 13   N1=4
      GO TO 6
 14   CONTINUE
 15   DO 16 I=1,100
      IF(AFBPS(I).EQ.0.)GO TO 17
      IF(AFBPS(I).EQ.-1.)GO TO 16
      KBP=KBP+1
      PQL(KBP+2*NPQ)=13.
      PQL(KBP)=-1.
      PQL(KBP+3*NPQ)=AFBPS(I)
 16   CONTINUE
 17   CONTINUE
      DO 18 I=1,4
      IF(BPS(I).EQ.5.)NP=NP+1
 18   CONTINUE
      TY=BPS(1)
      SUM=0.
      SUM1=0.
      TLNS=0.
      K=1
      IF(TY.EQ.5.)GO TO 30
      IF(TY.EQ.7.)GO TO 32
 19   I1C=0
      DO 20 I=1,10000
      IF(CONS(I).EQ.0.)GO TO 33
      IF(PORT.EQ.CONS(I+NC2))GO TO 21
      IF(PORT.EQ.CONS(I+2*NC2))GO TO 26
 20   CONTINUE
      GO TO 33
 21   NPC=2
 22   DO 23 I1=1,100
      IF(AFBPS(I1).EQ.0.)GO TO 25
      IF(PORT.EQ.AFBPS(I1))GO TO 24
 23   CONTINUE
      GO TO 25
 24   BLEG(LEG+NL)=PORT
      BLEG(LEG+3*NL)=SUM/SUM1
      BLEG(LEG+12*NL)=TLNS
      SUM=0.
      SUM1=0.
```

5.1.2.2 (Continued)

```
      TLNS=0.
      LEG=LEG+1
      BLEG(LEG)=PORT
      BEGM=1.
      AFBPS(I1)=-1.
       IF(I1C.EQ.1)GO TO 19
      IF(TY.EQ.3..OR.TY.EQ.6..OR.TY.EQ.10..OR.TY.EQ.31.
     1.OR.TY.EQ.33.)GO TO 45
   25 IF(NPC.EQ.2)GO TO 27
      GO TO 29
   26 NPC=3
      GO TO 22
   27 PORT=CONS(I+2*NC2)
   28 TY=CONS(I)
      CONS(I+NC2)=0.
      CONS(I+2*NC2)=0.
      CALL SCONST1(TLN,LEG,PORT,DIAO,DIAL,ALEN,I,NC2,TY,BLEG,
     1NPC,NL,CONNECT,BEGM)
      SUM=SUM+DIAL
      SUM1=SUM1+ALEN
      TLNS=TLNS+TLN
      GO TO 19
   29 PORT=CONS(I+NC2)
      GO TO 28
   30 PORT=BPS(K+2*NBP2)
      BPS(K+2*NBP2)=-1.
   31 BEGM=1.
      LEG=LEG+1
      BLEG(LEG)=PORT
      BLEG(LEG+2*NL)=K
      IBRLOC=BPS(K+5*NBP2)
      BRANCHP(IBRLOC+16*NBP2)=LEG
      BRANCHP(IBRLOC+20*NBP2)=-1.
      A=BRANCHP(IBRLOC+5*NBP2)
      IF(TY.EQ.7.)A=BRANCHP(IBRLOC+2*NBP2)
      TLEN=.1
      DIAO=5.
      CALL SPORT(TY,BEGM,LEG,A,DIAO,DIAL,TLEN,BLEG,NL)
      ALEN=TLEN
      SUM=SUM+DIAL
      SUM1=SUM1+ALEN
      TLN=.1
      TLNS=TLNS+TLN
      GO TO 19
   32 PORT=BPS(K+NBP2)
      BPS(K+NBP2)=-1.
      GO TO 31
      GO TO 19
   33 DO 35 I=1,10000
      IF(BPS(I).EQ.0.)GO TO 51
```

AD-A089 240 MCDONNELL AIRCRAFT CO ST LOUIS MO F/G 1/3
AIRCRAFT HYDRAULIC SYSTEMS DYNAMIC ANALYSIS. VOLUME VI. STEADY --ETC(U)
APR 80 R LEVEK, B YOUNG F33615-74-C-2016
UNCLASSIFIED AFAPL-TR-76-43-VOL-6 NL

2 of 4
AD
A089240


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

5.1.2.2 (Continued)

```
      DO 34 J=2,5
      IF(BPS(I+(J-1)*NBP2).EQ.PORT)GO TO 36
   34 CONTINUE
   35 CONTINUE
      GO TO 51
   36 TY=BPS(I)
      IBRLOC=BPS(I+5*NBP2)
      IF(TY.EQ.3..OR.TY.EQ.6..OR.TY.EQ.10..OR.TY.EQ.31.
     1.OR.TY.EQ.33.)GO TO 43
      BPS(I+(J-1)*NBP2)=-1.
      BLEG(LEG+NL)=PORT
      BLEG(LEG+4*NL)=I
      BEGM=0.
      CALL SCONST3(TLN,LEG,PORT,DIAO,DIAL,ALEN,J,TY,BLEG,NL,BEGM,
     1BRANCHP,NBP2,IBRLOC)
      SUM=SUM+DIAL
      SUM1=SUM1+ALEN
      TLNS=TLNS+TLN
      BLEG(LEG+3*NL)=SUM/SUM1
      BLEG(LEG+12*NL)=TLNS
      SUM=0.
      SUM1=0.
      TLNS=0.
      BRANCHP(IBRLOC+20*NBP2)=-1.
      IF(TY.EQ.9..OR.TY.EQ.24..OR.TY.EQ.25..OR.TY.EQ.91.)GO TO 37
      GO TO 38
   37 IF(TY.EQ.9.)NT=8+J
      IF(TY.EQ.24.)NT=5+J
      IF(TY.EQ.25.)NT=7+J
      IF(TY.EQ.91.)NT=6+J
      IF(TY.EQ.9..AND.J.EQ.4)BRANCHP(IBRLOC+13*NBP2)=-1.
      IF(TY.EQ.24.)BRANCHP(IBRLOC+(NT+3)*NBP2)=0.
      IF(TY.EQ.25.)BRANCHP(IBRLOC+(NT+4)*NBP2)=0.
      BRANCHP(IBRLOC+NT*NBP2)=LEG
   38 DO 40 K=1,10000
      M=0
      IF(BPS(K).EQ.0.)GO TO 53
      DO 39 L=2,5
      IF(BPS(K+(L-1)*NBP2).EQ.-1.)M=M+1
   39 CONTINUE
      DO 40 L=2,5
      IF(BPS(K+(L-1)*NBP2).GT.0..AND.M.GT.0.)GO TO 41
   40 CONTINUE
      GO TO 53
   41 PORT=BPS(K+(L-1)*NBP2)
      BPS(K+(L-1)*NBP2)=-1.
      IBRLOC=BPS(K+5*NBP2)
      TY=BPS(K)
      LEG=LEG+1
      BLEG(LEG)=PORT
```

5.1.2.2 (Continued)

```
      BLEG(LEG+2*NL)=K
      BEGM=1.
      CALL SCONST3(TLN,LEG,PORT,DIAO,DIAL,ALEN,L,TY,BLEG,NL,BEGM,
     1BRANCHP,NBP2,IBRLOC)
      BRANCHP(IBRLOC+20*NBP2)=-1.
      IF(TY.EQ.9..OR.TY.EQ.24..OR.TY.EQ.25..OR.TY.EQ.91.)GO TO 42
      GO TO 19
   42 IF(TY.EQ.9.)NT=8+L
      IF(TY.EQ.24.)NT=5+L
      IF(TY.EQ.25.)NT=7+L
      IF(TY.EQ.91.)NT=6+L
      IF(TY.EQ.9..AND.L.EQ.4)BRANCHP(IBRLOC+13*NBP2)=1.
      BRANCHP(IBRLOC+NT*NBP2)=LEG
      IF(TY.EQ.24.)BRANCHP(IBRLOC+(NT+3)*NBP2)=1.
      IF(TY.EQ.25.)BRANCHP(IBRLOC+(NT+4)*NBP2)=1.
      GO TO 19
   43 NPC=J
      DO 44 I1=1,100
      IF(AFBPS(I1).EQ.0.)GO TO 45
      IF(PORT.EQ.AFBPS(I1))GO TO 24
   44 CONTINUE
   45 BPS(I+(J-1)*NBP2)=-2.
      BRANCHP(IBRLOC+20*NBP2)=LEG
      CALL SCONST2(TLN,LEG,PORT,DIAO,DIAL,ALEN,IBRLOC,NBP2,TY,
     1BLEG,NL,BRANCHP,J)
      SUM=SUM+DIAL
      SUM1=SUM1+ALEN
      TLNS=TLNS+TLN
      IF(TY.EQ.10.)GO TO 46
      IF(TY.EQ.6.)NN=12
      IF(TY.EQ.3..OR.TY.EQ.33.)NN=6
      IF(TY.EQ.31.)NN=10
      BRANCHP(IBRLOC+NN*NBP2)=1.
      IF(J.EQ.3)BRANCHP(IBRLOC+NN*NBP2)=2.
   46 IF(J.GT.3)GO TO 47
      GO TO (47,47,48),J
   47 PORT=BPS(I+J*NBP2)
      BPS(I+J*NBP2)=-2.
      GO TO 49
   48 PORT=BPS(I+NBP2)
      BPS(I+NBP2)=-2.
   49 I1C=1
      NPC=J
      DO 50 I1=1,100
      IF(AFBPS(I1).EQ.0.)GO TO 19
      IF(PORT.EQ.AFBPS(I1))GO TO 24
   50 CONTINUE
      GO TO 19
   51 WRITE(6,52)PORT
   52 FORMAT(' ERROR---CANNOT FIND MATCH FOR PORT  ,F8.3)
```

5.1.2.2 (Continued)

```
      RETURN
   53 DO 55 K=1,10000
      IF(BPS(K).EQ.0.)GO TO 56
      TY=BPS(K)
      IF(BPS(K).EQ.7.)GO TO 54
      IF(BPS(K).NE.5.)GO TO 55
      IF(BPS(K+2*NBP2).EQ.-1.)GO TO 55
      GO TO 30
   54 IF(BPS(K+NBP2).EQ.-1.)GO TO 55
      GO TO 32
   55 CONTINUE
   56 DO 61 I=1,10000
      TY=BRANCHP(I)
      IF(TY.EQ.0.)GO TO 62
      IF(TY.EQ.4..OR.TY.EQ.8..OR.TY.EQ.34..OR.TY.EQ.35..OR.TY.EQ.36..OR
     1.TY.EQ.37..OR.TY.EQ.38.)GO TO 57
      GO TO 61
   57 LEG=LEG+1
      BLEG(LEG)=BRANCHP(I+NBP2)
      BLEG(LEG+NL)=BRANCHP(I+2*NBP2)
      DO 58 J=1,NPQ
      IF(PQL(J+2*NPQ).EQ.0.)GO TO 60
      IF(BLEG(LEG).EQ.PQL(J+3*NPQ).AND.BLEG(LEG+NL).EQ.PQL(J+1+3*NPQ))
     1GO TO 59
   58 CONTINUE
C       ****BUILD INTERNAL LEGS***
      GO TO 60
   59 TYL=PQL(J+NPQ)
      BLEG(LEG+2*NL)=TYL
      BLEG(LEG+4*NL)=TYL
      BRANCHP(I+15*NBP2)=LEG
   60 IF(TY.EQ.4..OR.TY.EQ.36..OR.TY.EQ.37..OR.TY.EQ.38.)GO TO 61
      IF(TY.EQ.8..AND.BRANCHP(I+3*NBP2).EQ.0.)GO TO 61
      LEG=LEG+1
      BLEG(LEG)=BRANCHP(I+NBP2)
      BLEG(LEG+NL)=BRANCHP(I+3*NBP2)
      BLEG(LEG+2*NL)=TYL
      BLEG(LEG+4*NL)=TYL
      BRANCHP(I+16*NBP2)=LEG
      IF(TY.EQ.8.)GO TO 61
      LEG=LEG+1
      BLEG(LEG)=BRANCHP(I+3*NBP2)
      BLEG(LEG+NL)=BRANCHP(I+2*NBP2)
      BLEG(LEG+2*NL)=TYL
      BLEG(LEG+4*NL)=TYL
      BRANCHP(I+17*NBP2)=LEG
      IF(TY.EQ.35.)GO TO 61
      LEG=LEG+1
      BLEG(LEG)=BRANCHP(I+NBP2)
      BLEG(LEG+NL)=BRANCHP(I+4*NBP2)
      BLEG(LEG+2*NL)=TYL
      BLEG(LEG+4*NL)=TYL
      BRANCHP(I+18*NBP2)=LEG
      LEG=LEG+1
      BLEG(LEG)=BRANCHP(I+4*NBP2)
      BLEG(LEG+NL)=BRANCHP(I+2*NBP2)
      BLEG(LEG+2*NL)=TYL
      BLEG(LEG+4*NL)=TYL
      BRANCHP(I+19*NBP2)=LEG
   61 CONTINUE
   62 CONTINUE
C       ****BUILD ILEP ARRAY***
      DO 65 I=1,10000
      IF(BLEG(I).EQ.0.)GO TO 66
      PM1=BLEG(I)
      PM2=BLEG(I+NL)
      BRP1=BLEG(I+2*NL)
      BRP2=BLEG(I+4*NL)
      DO 64 J=1,10000
      IF(PQL(J+2*NPQ).EQ.0.)GO TO 65
      DO 63 LL=3,6
      IF(PM1.EQ.PQL(J+NPQ*LL).AND.BRP1.EQ.PQL(J+NPQ))ILEP(I)=J
      ML=I
      IF(PM2.EQ.PQL(J+NPQ*LL).AND.BRP2.EQ.PQL(J+NPQ))ILEP(I+NL)=J
   63 CONTINUE
      IF(PM1.EQ PQL(J+NPQ*3).AND.PQL(J+2*NPQ).EQ.13.)ILEP(I)=J
      IF(PM2.EQ.PQL(J+NPQ*3).AND.PQL(J+2*NPQ).EQ.13.)ILEP(I+NL)=J
   64 CONTINUE
   65 CONTINUE
   66 CONTINUE
      DO 68 I=1,10000
      TY=PQL(I+2*NPQ)
      IF(TY.EQ.0.)GO TO 69
      IBLOC=PQL(I+NPQ)
      PQL(I+NPQ)=0.
      IF(TY.EQ.13..OR.TY.EQ.24..OR.TY.EQ.25.)GO TO 68
      BJCT=PQL(I+3*NPQ)
      IBRLOC=BPS(IBLOC+5*NBP2)
      N=2
      IF(TY.EQ.7.)N=1
      IF(TY.EQ.34.)N=4
      IF(TY.EQ.5..OR.TY.EQ.8..OR.TY.EQ.9..OR.TY.EQ.35.)N=3
      DO 67 J=1,N
      IF(BJCT.EQ.BRANCHP(IBRLOC+J*NBP2))BRANCHP(IBRLOC+(J+N)*NBP2)=I
   67 CONTINUE
   68 CONTINUE
   69 CONTINUE
      RETURN
      END
```

5.1.3 SCONST1, SCONST2 and SCONST3 Subroutines

Subroutines SCONST1, SCONST2 and SCONST3 are used to calculate static resistance coefficients for individual elements. SCONST1 is used for element types 1, 2, 11, 12, 21, 22, 23 and 32 which are all contained in CONS and CONNECT arrays, SCONST2 calculates the resistance coefficients for element types 3, 6, 10, 31 and 33. These elements are in BRANCHP and BPS array but are not branch point elements. SCONST3 calculates resistance coefficients for the remaining elements which are all branch point elements in BRANCHP and BPS arrays.

SCONST1 starts by routing the program flow to the proper calculation for the element type. The frictional resistance for the length of the element is calculated for each element. If there is a change in flow direction within the element, an energy loss coefficient (DPE1) is assigned or calculated. An equivalent orifice diameter is calculated for the type 32 restrictors and placed in BLEG array. The type 12 heat exchanger calculation determines the resistance coefficient for laminar and turbulent flow and places these in the appropriate BLEG columns. A change in port size from inlet to outlet with a change in passage size such as the reducer fitting energy loss is calculated in SPORT subroutine.

SCONST2 calculates the static resistance coefficients for the element type designated above, using the inlet and outlet passage lengths. The calculation is made through subroutine SPORT.

SCONST3 is similar to SCONST2 in that it calculates static resistance for the length of the port passage. Leg number data is also placed in BRANCHP array.

5.1.3.1 SCONST1 Variable Names

Variable	Description	Dimension
A	Inlet port size	IN or 16th IN
ALEN	Total length	IN
ALT	Altitude	FT
B	Outlet port size	IN or 16th IN
BEGM	Marker indicator for leg 0 = Begin leg 1 = End leg	--
BLEG	General prupose array	--
B1	Port size	IN or 16th IN
CD	Orifice discharge coefficient	--
CD1	Orifice discharge coefficient	--
CKL	Laminar flow coefficient	--
CKT	Turbulent flow coefficient	--
CONNECT	Static element data storage array	--
C2	Rated pressure drop	PSI
C3	Rated flow	GPM
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DEQ	Equivalent orifice diameter	IN
DIA	Port diameter	IN
DIAL	Sum of diameter x length	IN^2
DIAL1	Diameter x length inlet	IN^2
DIAL2	Diameter x length outlet	IN^2
DIAO	Previous element diameter	IN
DP	Rated pressure drop	PSI
DPE	Equivalent pressure drop	PSI

5.1.3.1 (Continued)

Variable	Description	Dimension
DPE1	Resistance Coefficient	--
DPL	Laminar pressure drop	PSI
DPT	Turbulent pressure drop	PSI
D1	Previous equivalent orifice diameter	IN
D100	Weight density at 100°F	LB/FT^3
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
I	Integer counter	--
LEG	Row number in BLEG array	--
NC2	Total rows in CONNECT array	--
NL	Total length of BLEG and ILEP arrays	--
NPC	Port indicator	--
OD	Tube outside diameter	IN
ORFD	Orifice diameter	IN
PAMB	Atmospheric ambient pressure	PSI
PORT	Junction number currently being investigated	--
Q	Rated flow	GPM
TEMP	Fluid temperature	°F
TLEN	Inlet port length	IN
TLEN1	Outlet port length	IN
TLN	Summation of inlet and outlet port lengths	IN
Ty	Element type	--
V	Rated viscosity	CENTISTOKES
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES

5.1.3.1 (Continued) - SCONST2 Variable Names

Variable	Description	Dimension
A	Inlet port size	IN or 16th IN
ALEN	Total length	IN
ALT	Altitude	FT
B	Outlet port size	IN or 16th IN
BLEG	General purpose array	--
BRANCHP	Dynamic element data storage array	--
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DIAL	Sum of diameter x length	IN^2
DIAL1	Diameter x length inlet	IN^2
DIAL2	Diameter x length outlet	IN^2
DIAO	Previous element diameter	IN
D100	Weight density at 100°F	LB/FT^3
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
IBRLOC	Row location of element in BRANCHP array	--
J	Indicator for inlet (2) or outlet (3) port	--
LEG	Row number in BLEG array	--
NBP2	Total length of BRONCHP array	--
NL	Total length of BLEG and ILEP arrays	--
PAMB	Atmospheric ambient pressure	PSI
PORT	Junction number currently being investigated	--
TEMP	Fluid temperature	°F
TLEN	Inlet port length	IN

5.1.3.1 (Continued)

Variable	Description	Dimension
TLEN1	Outlet port length	IN
TLN	Summation of inlet and outlet port lengths	IN
Ty	Element type	--
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
V100	Fluid viscosity at 100°F	CENTISTOKES

5.1.3.1 (Continued) - SCONST3 Variable Names

Variable	Description	Dimension
A	Inlet port size	IN or 16th IN
ALEN	Total length	IN
ALT	Altitude	FT
B	Outlet port size	IN or 16th IN
BEGM	Marker indicator for leg 0 = begin leg, 1 = end leg	--
BLEG	General purpose array	--
BRANCHP	Dynamic element data storage array	--
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DIAL	Sum of diameter x length	IN^2
DIAL1	Diameter x length inlet	IN^2
DIAL2	Diameter x length outlet	IN^2
DIAO	Previous element diameter	IN
D100	Weight density at 100°F	LB/FT^3
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FKYUDJ	Viscosity-pressure correction factor at fluid temperature	--
IBRLOC	Row location of element in BRANCHP array	--
J	Indicator for inlet (2) or outlet (3) port	--
LEG	Row number in BLEG array	--
NBP2	Total length of BRANCHP array	--
NL	Total length of BLEG and ILEP arrays	--
N1	Assembly indicator to indicate which port first assembled	--
N2	Inlet (3) and outlet (4) port indicator	--
PAMB	Atmospheric ambient pressure	PSI

5.1.3.1 (Continued)

Variable	Description	Dimension
PORT	Junction number currently being investigated	--
TEMP	Fluid temperature	°F
TLEN	Inlet port length	IN
TLEN1	Outlet port length	IN
TLN	Summation of inlet and outlet port lengths	IN
Ty	Element type	--
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
V100	Fluid viscosity at 100°F	CENTISTOKES

5.1.3.2 SCONST1 , SCONST2 and SCONST3 Subroutine Listing

```
      SUBROUTINE SCONST1(TLN,LEG,PORT,DIAO,DIAL,ALEN,I,NC2,TY,BLEG,NPC
     1,NL,CONNECT,BEGM)
C           RESISTANCE CONSTANTS FOR TUBE     REV 9/20/79
      DIMENSION BLEG(1),CONNECT(1)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB ALT,FLUIDF,FLUIDK,V100
      IF(TY.EQ.21.)GO TO 2
      IF(TY.EQ.22.)GO TO 6
      IF(TY.EQ.2..OR.TY.EQ.23..OR.TY.EQ.32..OR.TY.EQ.12.)GO TO 7
      IF(TY.EQ.11.)GO TO 8
      OD=CONNECT(I+3*NC2)
      IF(OD.GE.4.)OD=OD/16.
      DIA=OD-(2.*CONNECT(I+4*NC2))
      TLEN=CONNECT(I+5*NC2)
      A=DIA
    1 CALL SPORT(TY,BEGM,LEG,A,DIAO,DIAL,TLEN,BLEG,NL)
      ALEN=TLEN
      TLN=TLEN
      DPE1=CONNECT(I+6*NC2)
      GO TO 5
C          RESISTANCE CONSTANTS FOR 45 DEG ELBOW      REV 9/20/79
    2 DPE1=.2179
    3 A=CONNECT(I+3*NC2)
      B=CONNECT(I+4*NC2)
      IF(NPC.EQ.2)GO TO 4
      B1=A
      A=B
      B=B1
    4 TLEN=3.*A
      CALL SPORT(TY,BEGM,LEG,A,DIAO DIAL1,TLEN,BLEG,NL)
      TLEN1=3.*B
      CALL SPORT(TY,BEGM,LEG,B,DIAO,DIAL2,TLEN1,BLEG,NL)
      TLN=3.*A+3.*B
      ALEN=TLEN+TLEN1
      DIAL=DIAL1+DIAL2
    5 DPE=DPE1*((1.799E-5*D100)/(A**4))
      BLEG(LEG+10*NL)=BLEG(LEG+10*NL)+DPE
      IF(TY.EQ.32.)GO TO 9
      IF(TY.EQ.12.)GO TO 12
      RETURN
C          RESISTANCE CONSTANTS FOR 90 DEG ELBOW      REV 9/20/79
    6 DPE1=1.2
      GO TO 3
C     RESISTANCE CONSTANTS FOR REDUCER FTG & UNION       REV 9/20/79
    7 DPE1=0.
      GO TO 3
C          RESISTANCE CONSTANTS FOR HOSE          REV 9/20/79
    8 A=CONNECT(I+4*NC2)
      TLEN=CONNECT(I+5*NC2)
      GO TO 1
```

5.1.3.2 (Continued)

```
C        RESISTANCE CONSTANTS FOR 2 WAY RESTRICTOR    REV 9/20/79
    9 ORFD=CONNECT(I+5*NC2)
      CD1=CONNECT(I+6*NC2)
      IF(ORFD.EQ.0.)ORFD=.00001
      IF(ORFD.GT..9)GO TO 11
      CD=CD1/((1.-(ORFD/CONNECT(I+3*NC2))**4)**.5)
   10 DPE=D100/((236.*(ORFD**2)*CD)**2)
      BLEG(LEG+10*NL)=BLEG(LEG+10*NL)+DPE
      D1=BLEG(LEG+6*NL)
      DEQ=ORFD
      IF(D1.GT.0.)DEQ=((D1**4*ORFD**4)/(D1**4+ORFD**4))**.25
      BLEG(LEG+6*NL)=DEQ
      RETURN
   11 C2=CONNECT(I+5*NC2)
      C3=CONNECT(I+6*NC2)
      ORFD=(C3/(236.*.6*((C2/D100)**.5)))**.5
      CD=.6
      GO TO 10
C           RESISTANCE CONSTANTS FOR HEAT EXCH     REV   9/20/79
   12 DP=CONNECT(I+5*NC2)
      Q=CONNECT(I+6*NC2)
      V=CONNECT(I+7*NC2)
      CKL=DP/Q
      CKT=DP/(Q**1.75)
      DPL=CKL*(V100/V)
      DPT=CKT*((V100/V)**.25)
      BLEG(LEG+8*NL)=BLEG(LEG+8*NL)+DPL
      BLEG(LEG+9*NL)=BLEG(LEG+9*NL)+DPT
      RETURN
      END
```

5.1.3.2 (Continued) (SCONST2)

```
      SUBROUTINE SCONST2(TLN,LEG,PORT,DIAO,DIAL,ALEN,IBRLOC,NBP2,TY,
     ABLEG,NL,BRANCHP,J)
      DIMENSION BRANCHP(1),BLEG(1)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      N2=4
      IF(J.EQ.2)N2=3
      IF(TY.EQ.10.)GO TO 2
      N1=6
      IF(TY.EQ.31.)N1=10.
    1 BRANCHP(IBRLOC+N1*NBP2)=1.
      IF(J.EQ.3)BRANCHP(IBRLOC+N1*NBP2)=2.
    2 A=BRANCHP(IBRLOC+3*NBP2)
      B=BRANCHP(IBRLOC+4*NBP2)
      TLEN=3.*A
      TLEN1=3.*B
      CALL SPORT(TY,BEGM,LEG,A,DIAO,DIAL1,TLEN,BLEG,NL)
      BRANCHP(IBRLOC+3*NBP2)=A
      CALL SPORT(TY,BEGM,LEG,B,DIAO,DIAL2,TLEN,BLEG,NL)
      BRANCHP(IBRLOC+4*NBP2)=B
      DIAO=BRANCHP(IBRLOC+N2*NBP2)
      TLN=3.*A+3.*B
      ALEN=TLEN+TLEN1
      DIAL=DIAL1+DIAL2
      RETURN
      END
```

5.1.3.2 (Continued) (SCONST3)

```
      SUBROUTINE SCONST3(TLN,LEG,PORT,DIAO,DIAL,ALEN,J,TY,BLEG,NL,BEGM,
     1BRANCHP,NBP2,IBRLOC)
      DIMENSION BLEG(1),BRANCHP(1)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      IF(TY.EQ.7.)GO TO 1
      IF(TY.EQ.24..OR.TY.EQ.5..OR.TY.EQ.8..OR.TY.EQ.9..OR.TY.EQ.35.)GO T
     1O 3
      IF(TY.EQ.25..OR.TY.EQ.34..OR.TY.EQ.36.)GO TO 4
      IF(TY.EQ.4..OR.TY.EQ.37..OR.TY.EQ.38..OR.TY.EQ.91..OR.TY.EQ.92.)GO
     1 TO 5
    1 A=BRANCHP(IBRLOC+2*NBP2)
    2 TLEN=3.*A
      CALL SPORT(TY,BEGM,LEG,A,DIAO,DIAL1,TLEN,BLEG,NL)
      IF(TY.EQ.24.)BRANCHP(IBRLOC+(J+2)*NBP2)=A
      IF(TY.EQ.25.)BRANCHP(IBRLOC+(J+3)*NBP2)=A
      ALEN=TLEN
      TLN=3.*A
      DIAL=DIAL1
      IF(TY.EQ.4..OR.TY.EQ.5..OR.TY.EQ.7..OR.TY.EQ.8..OR.TY.EQ.9.
     1.OR.TY.EQ.91..OR.TY.EQ.92.)GO TO 6
      RETURN
    3 A=BRANCHP(IBRLOC+(J+2)*NBP2)
      IF(TY.EQ.5..AND.J.EQ.3)BRANCHP(IBRLOC+16*NBP2)=LEG
      GO TO 2
    4 A=BRANCHP(IBRLOC+(J+3)*NBP2)
      GO TO 2
    5 A=BRANCHP(IBRLOC+(J+1)*NBP2)
      GO TO 2
    6 IF(BEGM.EQ.0.)RETURN
      TLEN1=.0001
      B=3.95
      CALL SPORT(TY,BEGM,LEG,B,DIAO,DIAL2,TLEN,BLEG,NL)
      ALEN=TLEN+TLEN1
      DIAL=DIAL1+DIAL2
      RETURN
    7 CONTINUE
      RETURN
      END
```

5.1.4 SPORT Subroutine

SPORT is called through the SCONST1, SCONST2 and SCONST3 subroutines for the primary purpose of calculating the static fluid resistance coefficient for line and hose lengths, fitting and port lengths and changes in cross-section. Values of resistance are summed directly into BLEG array. Figure 30 shows the general flow chart for the subroutine.

5.1.4.1 Description of Operation

Subroutine SPORT performs three tasks:

1) Converts port sizes to internal diameters if data is input using size option.
2) Calculates static resistance coefficients using Functions SFRICL, SFRICT, SKBS and SKSB.
3) Returns Diameter times Length term to BUILD subroutine for use in calculating the average diameter of the leg.

First, a test is made to determine if the port size convention used is the actual inside diameter or the equivalent tube size. If the input size value is greater than zero and less than 3.999, the actual input value for size is used. Input equivalent tube port sizes of 4.0 and greater may be used.

Calculation of the laminar and turbulent frictional coefficients are made through SFRICL and SFRICT functions respectively. Next, the local energy loss is calculated with either SKBS or SKSB function. The laminar frictional coefficient is summed into column 9 of BLEG array. The turbulent frictional coefficient is summed into column 10 and the local energy loss is summed into column 11.

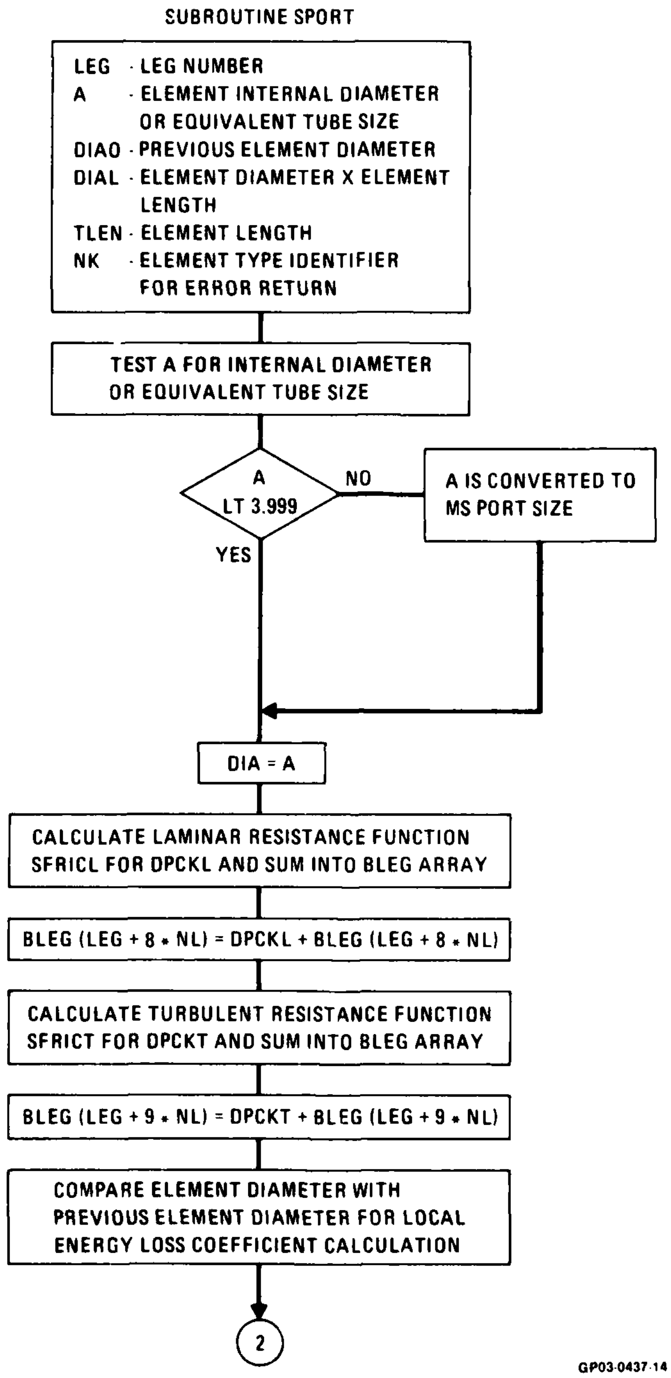

SUBROUTINE SPORT
LEG - LEG NUMBER
A - ELEMENT INTERNAL DIAMETER OR EQUIVALENT TUBE SIZE
DIAO - PREVIOUS ELEMENT DIAMETER
DIAL - ELEMENT DIAMETER X ELEMENT LENGTH
TLEN - ELEMENT LENGTH
NK - ELEMENT TYPE IDENTIFIER FOR ERROR RETURN
TEST A FOR INTERNAL DIAMETER OR EQUIVALENT TUBE SIZE
A LT 3.999
NO
A IS CONVERTED TO MS PORT SIZE
YES
DIA = A
CALCULATE LAMINAR RESISTANCE FUNCTION SFRICL FOR DPCKL AND SUM INTO BLEG ARRAY
BLEG (LEG + 8 * NL) = DPCKL + BLEG (LEG + 8 * NL)
CALCULATE TURBULENT RESISTANCE FUNCTION SFRICT FOR DPCKT AND SUM INTO BLEG ARRAY
BLEG (LEG + 9 * NL) = DPCKT + BLEG (LEG + 9 * NL)
COMPARE ELEMENT DIAMETER WITH PREVIOUS ELEMENT DIAMETER FOR LOCAL ENERGY LOSS COEFFICIENT CALCULATION
2
GP03-0437-14

FIGURE 30
SPORT SUBROUTINE GENERAL FLOW CHART

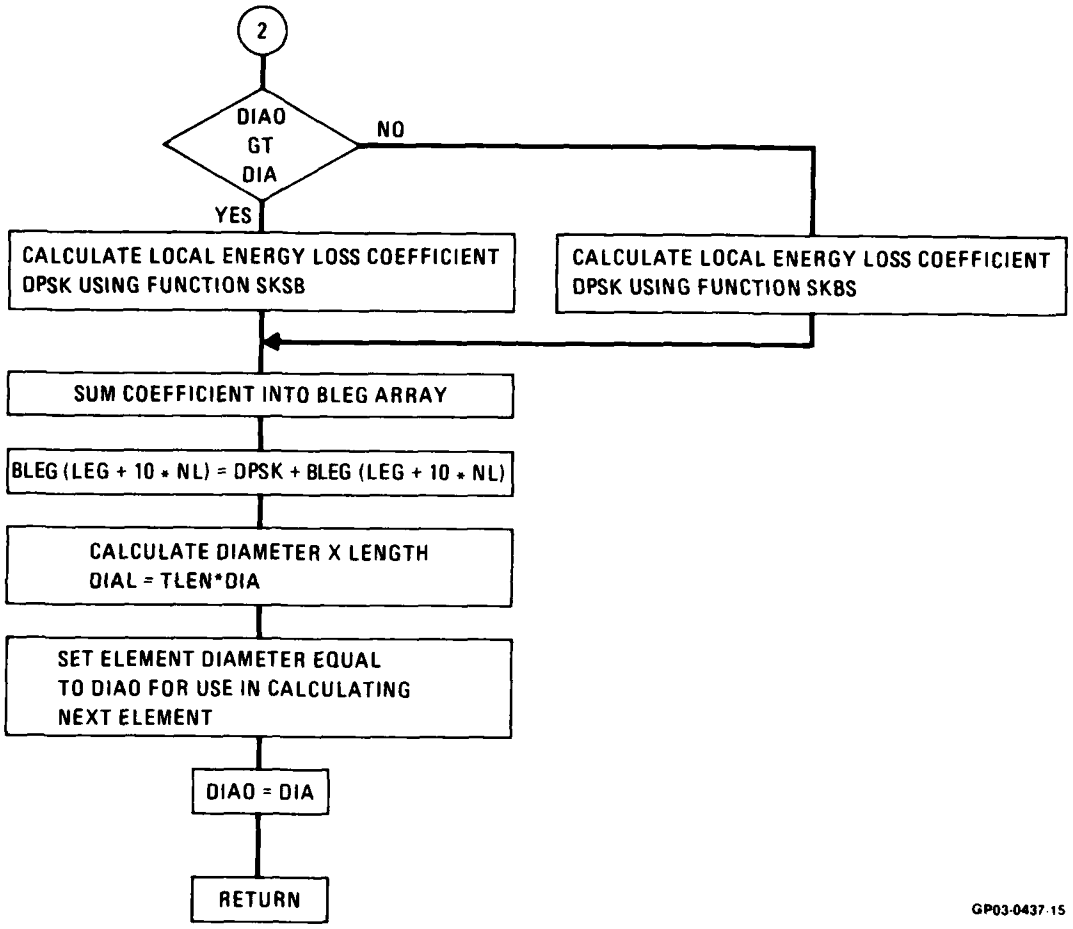

2
DIAO
GT
DIA
NO
YES
CALCULATE LOCAL ENERGY LOSS COEFFICIENT DPSK USING FUNCTION SKSB
CALCULATE LOCAL ENERGY LOSS COEFFICIENT DPSK USING FUNCTION SKBS
SUM COEFFICIENT INTO BLEG ARRAY
BLEG (LEG + 10 * NL) = DPSK + BLEG (LEG + 10 * NL)
CALCULATE DIAMETER X LENGTH
DIAL = TLEN*DIA
SET ELEMENT DIAMETER EQUAL TO DIAO FOR USE IN CALCULATING NEXT ELEMENT
DIAO = DIA
RETURN
GP03-0437-15

FIGURE 30 (Continued)

A calculation of diameter times length (DIAL) is next made and the old diameter (DIAO) is then set to the current element diameter for use in calculating the next element local energy loss coefficient.

5.1.4.2 SPORT Variable Names

Variable	Description	Dimension
A	Inlet port size	IN or 16th IN
AA	Inlet port size	IN
AB	Temporary inlet port size	IN or 16th IN
ALT	Altitude	FT
BEGM	Marker Indicator for Leg 0 = begin leg; 1 = end leg	--
BLEG	General purpose array	--
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DIA	Port diameter	IN
DIAL	Sum of diameter x length	IN^2
DIAO	Previous element diameter	IN
DPCKL	Laminar pressure drop coefficient	--
DPCKT	Turbulent pressure drop coefficient	--
DPSK	Pressure drop coefficient due to change in section size	--
D100	Weight density at 100°F	LB/FT^3
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
LEG	Row number in BLEG array	--
NL	Total length of BLEG and ILEP arrays	--
PAMB	Atmospheric ambient pressure	PSI
SZ	Inlet port size	IN
SZ1	Intermediate calculation of port size	--
SZ2	Intermediate calculation of port size	--
TEMP	Fluid temperature	°F

5.1.4.2 (Continued)

Variables	Description	Dimensions
TLEN	Inlet port length	IN
TY	Element type	--
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
V100	Fluid viscosity at 100°F	CENTISTOKES

5.1.4.3 SPORT Subroutine Listing

```
      SUBROUTINE SPORT(TY,BEGM,LEG,A,DIAO,DIAL,TLEN,BLEG,NL)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION BLEG(1)
      IF(A.LT.3.999)GO TO 4
      SZ=0.
      SZ1=0.
      SZ2=0.
      IF(A.LT.7.)GO TO 3
      IF(A.LT.13.)GO TO 1
      AB=A
      IF(A.GT.32.)AB=32.
      SZ2=(AB-12.)*.004
      IF(AB.GT.20.)SZ2=SZ2-.001
      IF(AB.GT.16.)SZ2=SZ2-.001
      AA=10.
      GO TO 2
    1 AA=A
      IF(A.EQ.11..OR.A.EQ.12.)AA=10.
    2 SZ1=(AA-6.)*.016
      IF(AA.GT.7.)SZ1=SZ1-.001
    3 SZ=(A/16.)-(.078+SZ1+SZ2)
      TLEN=(TLEN/A)*SZ
      A=SZ
    4 DIA=A
      IF(BEGM.EQ.0.)GO TO 5
      IF(TY.EQ.4..OR.TY.EQ.5..OR.TY.EQ.7..OR.TY.EQ.8..OR.TY.EQ.9.
     1.OR.TY.EQ.91..OR.TY.EQ.92.)DIAO=5.
      IF(TY.EQ.24..OR.TY.EQ.25..OR.TY.EQ.34..OR.TY.EQ.35..OR.TY.EQ.36.
     1.OR.TY.EQ.37..OR.TY.EQ.38.)DIAO=A
    5 DPCKL=SFRICL(V100,TLEN,D100,DIA)
      BLEG(LEG+8*NL)=DPCKL+BLEG(LEG+8*NL)
      DPCKT=SFRICT(V100,TLEN,D100,DIA)
      BLEG(LEG+9*NL)=DPCKT+BLEG(LEG+9*NL)
      IF(DIAO.LE.0.)GO TO 6
      IF(DIAO.GT.DIA)GO TO 7
      DPSK=SKSB(DIAO,DIA,D100)
      GO TO 8
    6 DPSK=0.
      GO TO 8
    7 DPSK=SKBS(DIAO,DIA,D100)
    8 CONTINUE
      BLEG(LEG+10*NL)=DPSK+BLEG(LEG+10*NL)
      DIAL=TLEN*DIA
      DIAO=DIA
      A=DIA
      RETURN
      END
```

5.2 CALCULATION OF STATIC FLUID RESISTANCE COEFFICIENTS

Two types of fluid losses are considered in SSFAN. These are static fluid resistance losses and dynamic losses. Dynamic losses are those for which the resistance factor is calculated during the solution procedure because the resistance equation used is dependent on flow magnitude flow direction or pressure drop. Dynamic losses are discussed in section 6.

The fluid losses in the course of the motion of a fluid are due to the irreversible transformation of mechanical energy into heat. This energy transformation is due to the molecular and turbulent viscosity of the moving medium.

There exist two different types of static fluid losses: 1) the frictional losses; ΔP_{fr} and 2) the local energy losses ΔP_1

The frictional losses are due to the viscosity (molecular and turbulent) of the fluids, which manifests itself during their motion and is a result of the exchange of momentum between molecules at laminar flow and between individual particles of adjacent fluid layers moving at different velocities, at turbulent flow. These losses take place along the entire length of the pipe.

The local energy losses appear at a disturbance of the normal flow of the stream, such as its separation from the wall and the formation of eddies at places of alteration of the pipe configuration where a tube and a fitting join and at obstacles in the pipe. The losses of dynamic pressure occurring with the discharge of the stream from a pipe into a large volume such as a reservoir must also be classed as local energy losses.

The phenomenon of flow separation and eddy formation is linked with the difference between the flow velocities in the cross section, and with a positive pressure gradient along the stream, which appears when the motion is slowed down in an expanding channel, in accordance with Bernoulli's equation. The difference between the velocities in the cross section at a negative pressure gradient does not lead to flow separation. The flow in smoothly converging stretches is even more stable than in stretches of constant section.

Static fluid resistance coefficients are calculated for all elements through the SCONST1, SCONST2 and SCONST3 subroutines.

These subroutines utilize the functions SFRICL, SFRICT, SKBS and SKSB through SPORT subroutine.

5.2.1 Static Resistance Functions

There are four static resistance functions. Function SFRICL calculates a normalized pressure loss or resistance coefficient for laminar flow due to the length of the element. Function SFRICT calculates the turbulent coefficient for the same element.

Function SKSB calculates the normalized local energy loss coefficient for flow from a small diameter element to a larger diameter element. Function SKBS calculates the normalized local energy loss coefficient for flow from a large diameter element to a smaller diameter element.

5.2.1.1 Description of Function Usage

The SFRICL, SFRICT, SKSB and SKBS functions generally are called from SPORT subroutine. SFRICL and SFRIC utilize values for atmospheric viscosity and density at 100°F, along with the passage diameter and the passage length. The laminar and turbulent reference pressure drops or static resistance coefficients are calculated for a flow of one gpm and passed back to the calling subroutine.

SKSB and SKBS require the diameter of the previous assembled element for calculation of the local energy loss. The atmospheric fluid density at 100°F is required, but not viscosity since the local energy loss is independent of viscosity. These local resistance energy losses are referenced to one gpm flow.

5.2.1.2 Assumptions

The piping system is assumed to be circular and flow one-dimensional for the fluid resistance calculations. Local energy losses are assumed to occur at all connecting elements where there is a difference between the passage diameters of the two elements.

5.2.1.3 Computations

Frictional Losses - The Darcy equation for head loss in circular pipes may be derived by dimensional analysis or similitude and may be written in the form of:

$$h_L = f \frac{L}{D} \frac{V^2}{2g} \qquad (1)$$

where:	h_L -	head loss in feet
	f -	friction factor
	L -	length of passage in feet
	D -	internal passage diameter in feet
	V -	fluid velocity in feet/second
	g -	acceleration of gravity - 32.2 feet per second per second

Using appropriate dimensional conversion factors, this equation may be rewritten (Reference 3) as:

$$\Delta P = 1.799 \times 10^{-5} \; f \; \frac{l}{d} \; \rho \frac{Q^2}{d4} \qquad (2)$$

Δ P --	pressure drop psi
f -	friction factor
l -	length of passage in inches
d -	internal passage diameter in inches
ρ -	weight density of the fluid in pounds per cubic feet
Q -	flow rate in gallons per minute

Reynolds Number (Re) is a dimensionless ratio of the fluid inertia forces to the viscous forces where:

$$Re = \frac{\rho_M V d}{u} = \frac{Vd}{\nu}$$

u	-	absolute viscosity
ν	-	kinematic viscosity
ρ_M	-	mass density
d	-	internal diameter
V	-	fluid velocity

The friction factors in SSFAN are defined as a function of Reynolds Number for laminar, transition and turbulent flows; therefore Reynolds Number is most conveniently defined in terms with

Q	-	gpm
ν	-	centipoise
d	-	inches

With $Re = \frac{Vd}{\nu}$ (3)

V	-	in per second
d	-	in
ν	-	in^2 per second

Conversion to Q in terms of V is as follows:

$$Q = \frac{60}{231} \times \frac{\pi}{4} d^2 \times V$$

or $$V = \frac{231}{60} \times \frac{4}{\pi} \times \frac{Q}{d^2}$$

$$V = 4.902 \frac{Q}{d^2} \qquad (4)$$

The conversion of kinematic viscosity from in^2/sec to centistokes is

$$1 \text{ (centistoke)} = 1.55 \times 10^{-3} \text{ (in}^2\text{/sec)} \qquad (5)$$

or $1 \text{ (in}^2\text{/sec)} = 645 \text{ (centistokes)}$

Substituting equations (4) and (5) into (3) yields

$$Re = 3161.77 \frac{Q}{d\nu} \qquad (6)$$

with	Q - gpm
	d - in
	ν - centistokes

The above equations are used in deriving the function equations.

Function SFRICL

SFRICL is a normalized equation for calculating the laminar frictional pressure drop for a given passage length of constant cross-section. Using a friction factor for laminar flow (Reference 5) of

$$f = \frac{64}{Re}$$

and substituting equation (6) for Re in the friction factor term above,

$$f = 2.024 \times 10^{-2} \times \frac{\nu}{Qd} \qquad (7)$$

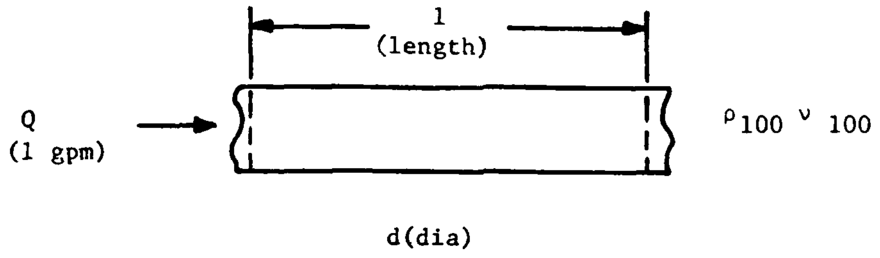
Substituting equation (7) into equation (2) yields

$$\Delta P = 3.64 \times 10^{-7} \times \frac{1}{d} \times \frac{\rho Q^2}{d^4} \times \frac{d\nu}{Q}$$

or

$$\Delta P = 3.64 \times 10^{-7} \times \frac{\rho Q \ \nu 1}{d^4} \qquad (8)$$

To normalize the above equation for flow, viscosity and density, flow is given a value of 1 gpm, viscosity is referenced to a viscosity at atmospheric pressure and 100°F and density is referenced to density at atmospheric pressure and 100°F, see Figure 31.



Laminar Flow Resistance

FIGURE 31

The laminar resistance coefficient or normalized laminar pressure drop may now be defined as K_9 where the 9 indicates the column location of the laminar resistance coefficients in BLEG array.

Equation (8) is now written in terms of the above reference viscosity, density, and flow to be

$$SFRICL = K_9 = \Delta P_{100(REF)} = 3.64 \times 10^{-7} \times \frac{\rho_{100}\ \nu_{100}\ l \times 1}{d^4} \qquad (9)$$

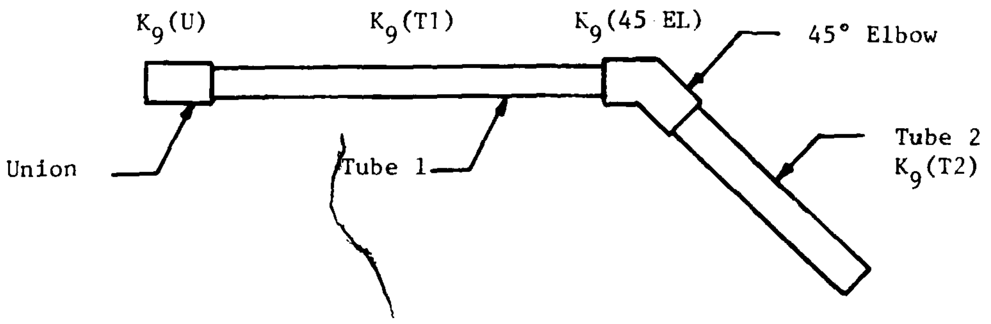
ρ_{100} - weight density in lb/per cubic feet at atmospheric pressure and 100°F

ν_{100} - kinematic viscosity in centistokes at atmospheric pressure and 100°F

l - length of passage in inches

d - internal passage diameter in inches

The individual element K_9 values are summed for each leg since series resistances are additive, see Figure 32.



$$K_9\ (LEG) = \sum_{\text{Elements}} K_9$$

$$K_9\ (LEG) = K_9(U) + K_9(T1) + K_9(45\ EL) + K_9(T2)$$

Example for Summation of K_9 Values

FIGURE 32

Using 100°F as a reference temperature ensures that all data is referenced to a common reference point. It should be noted that some element flow pressure drop data may be input for viscosities at temperatures other than 100°F, but these data are converted to the 100°F reference point within the specific subroutine.

Examination of equation (9) will show that when the K_9 values of BLEG array are multiplied by the factor

$$\frac{\rho_p}{\rho_{100}} \times \frac{\nu_p}{\nu_{100}} \times Q_{LEG}$$

ρ_p - density corrected for temperature and pressure

ν_p - viscosity corrected for temperature and pressure

The true ΔP will be calculated using the corrected viscosity and density at pressure and temperature. This correction is made in Subroutine TTL and is described more fully there. Equation (9) when multiplied by the above factor will result in

$$\Delta P_{LAMINAR\ FRIC} = 3.64 \times 10^{-7} \times \frac{\rho_p\ Q_{LEG}\ \nu_p\ l}{d^4} \qquad (10)$$

Function SFRICT

SFRICT is a normalized equation for calculating the turbulent pressure drop for a given passage length of constant cross-section.

Using the Blasius friction factor for turbulent flow (Reference 5),

$$f = \frac{.316}{(Re)^{.25}}$$

and substituting equation (6) for Re in the term above,

$$f = 4.214 \times 10^{-2} \quad \frac{d^{.25}\ \nu^{.25}}{Q^{.25}}$$

Substituting this value of friction factor into equation (2) yields

$$\Delta P = 7.591 \times 10^{-7} \quad \frac{d^{.25}\ \nu^{.25}}{Q^{.25}} \quad \frac{1}{d} \quad \frac{\rho\ Q^{2}}{d^{4}}$$

or when terms are combined

$$\Delta P = 7.591 \times 10^{-7} \quad \frac{\rho 1\ \nu^{\ .25}\ Q^{1.75}}{d^{4.75}} \qquad (11)$$

The normalized form of the equation is now found similar to SFRICL where density and viscosity are referenced to atmospheric pressure and 100°F with Q equal to 1 gpm.

The turbulent resistance coefficient or normalized turbulent pressure drop may now be defined as K_{10} where 10 indicates the column location of the turbulent resistance coefficients in BLEG array.

$$\text{SFRICT} = K_{10} = \Delta P_{100(\text{REF})} = 7.591 \times 10^{-7} \frac{\rho_{100}\ \nu_{100}\ 1}{d^{4.75}} \qquad (12)$$

with dimensions and designations as for equation (9) above.

Individual element K_{10} values are summed to give a total of

$$K_{10}\ (\text{LEG}) = \sum_{\text{ELEMENTS}} K_{10}$$

When the BLEG K_{10} values are used by TTL for calculation, the K_{10} values (equation (12)) are multiplied by

$$\frac{\rho_p}{\rho_{100}} \times \frac{\nu_p}{\nu_{100}}^{.25} \times Q^{\ 1.75}$$

This then gives the actual ΔP corrected for viscosity and density at temperature and pressure.

$$\Delta P_{\text{TURBULENT FRIC}} = 7.591 \times 10^{-7} \frac{\rho_p \, \nu_p^{.25} \, l \, Q^{1.75}}{d^{4.75}} \tag{13}$$

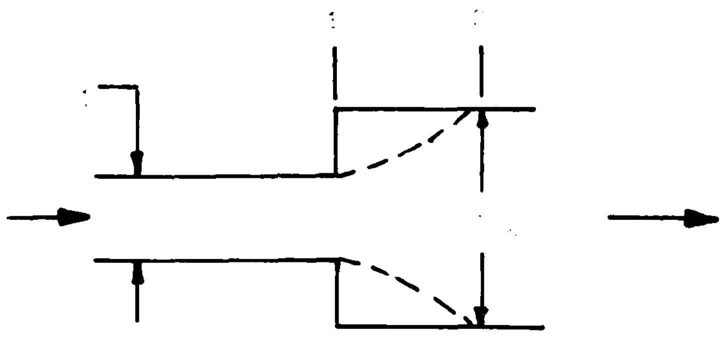
<u>LOCAL ENERGY LOSSES</u>

Losses which occur in a system of connected pipes and fittings due to the change in diameter from one element to the next are termed local energy losses. These are of two types: (1) sudden expansion and (2) sudden contraction.

The case of sudden expansion (see Figure 33) is one in which the momentum equation may be applied together with the Bernoulli equation to obtain an energy loss expression in terms of velocities, (Reference 5). This may be written in the form similar to the Darcy equation where:

$$P_1A_2 - P_2A_2 = \frac{Q\rho}{g} (V_2 - V_1)$$

P_1	-	Pressure at 1
P_2	-	Pressure at 2
A_2	-	Area at 2
Q	-	Flow
ρ	-	Density
V_1	-	Fluid velocity at 1
V_2	-	Fluid velocity at 2
g	-	Gravitational constant



Sudden Expansion

FIGURE 33

Bernoulli's equation written between sections 1 and 2 with the sudden expansion loss is

$$\frac{V_1^2}{2g} + \frac{P_1}{\rho} = \frac{V_2^2}{2g} + \frac{P_2}{\rho} + he$$

where he - sudden expansion head loss

solving for $(P_1 - P_2) / \rho$ in each equation and equating

$$\frac{P_1 - P_2}{\rho} = \frac{Q}{A_2 g} (V_2 - V_1) = \frac{V_2^2 - V_1^2}{2g} + he$$

Since $Q/A_2 = V_2$,

$$he = \frac{2V_2(V_2 - V_1)}{2g} - \frac{V_2^2 - V_1^2}{2g} - \frac{(V_1 - V_2)^2}{2g}$$

From the Darcy equation

$$he = K_E \frac{V_1^{\ 2}}{2g} = \left[1 - \left(\frac{D_1}{D_2}\right)^2\right]^2 \frac{V_1^{\ 2}}{2g}$$

where K_E is $$K_E = 1 - \left[\left(\frac{D1}{D2}\right)^2\right]^2 \qquad (14)$$

Figure 34 shows a graph of this equation.

For a loss due to sudden contraction, see Figure 35.

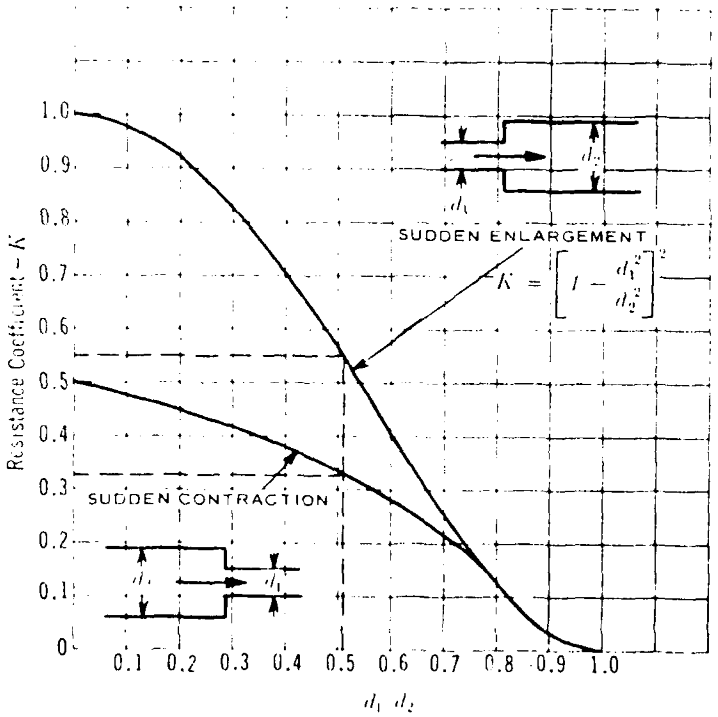
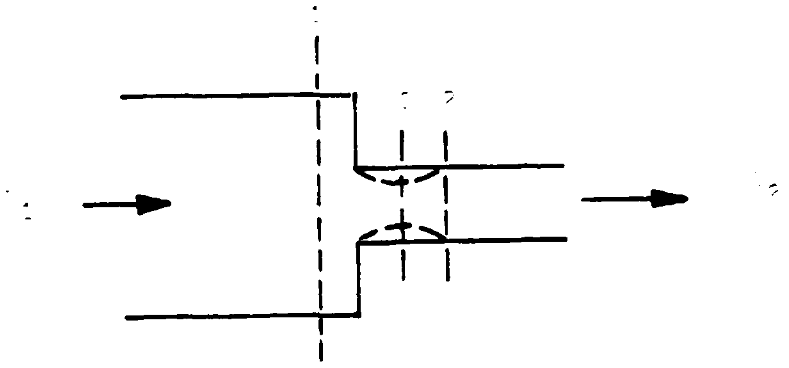


FIGURE 34

Resistance Due to Sudden Enlargements and Contractions



Sudden ontraction

FIGURE 35

The same analysis may be applied, provided that the amount of contraction is known. The process of converting pressure energy into velocity energy is very efficient. Therefore, the loss from 1 to 0 is small compared with the reconversion of kinetic energy back to pressure energy from 0 to 2. Applying the above equations, the expansion for the section from 0 to 2 is

$$hc = \frac{(V_0 - V_2)^2}{2g}$$

From the continuity equation

$$V_0 C_c A_2 = V_2 A_2$$

C_c - contraction coefficient

substituting

$$h_c = \left(\frac{1}{C_c} - 1\right)^2 \times \frac{V_2^2}{2g}$$

For a sharp edged opening the term

$$K_c = \left(\frac{1}{C_c} - 1\right)^2$$

has experimentally been shown to be

$$K_c = .5 \left[1 - \left(\frac{D_1}{D_2}\right)^2\right] \qquad (15)$$

Figure 5-12 shows this equation in graph form

Function SKSB

SKSB is a normalized function for an energy loss when flow is from a small passage to a larger passage.

With K_E replacing $f \frac{1}{d}$ in equation (2)

$$\underset{\text{Expansion}}{\Delta P_{100}} = 1.799 \times 10^{-5} \; K_E \; \frac{\rho_{100} Q_1^{\,2}}{d^4}$$

K_E - expansion energy loss coefficient

ρ_{100} - fluid weight density at 100°F(lb/ft^3)

Q_1 - flow in gpm (1 gpm)

d - upstream passage diameter in inches

SKSB calculates the normalized ΔP for a flow of 1 gpm and a fluid density at 100°F. These energy loss terms are calculated individually for applicable elements and summed for the entire leg where it is placed into column 11 of BLEG array.

When the normalized ΔP is used by subroutine TTL, it is corrected for actual flow and density as follows:

$$\underset{\text{energy loss}}{\Delta P_{\text{corrected}}} = \underset{\text{Expansion}}{\Delta P_{100}} \times \frac{\rho_p}{\rho_{100}} \times Q^2$$

ρ_p - fluid weight density corrected for pressure

Q - Actual flow in leg in gpm

<u>Function SKBS</u>

SKBS is a normalized function for an energy loss when flow is from a large diameter passage to a smaller diameter passage.

With K_c replacing $f \frac{1}{d}$ in equation (2),

$$\underset{\text{Contraction}}{\Delta P100} = 1.799 \times 10^{-5} \; K_c \; \frac{\rho_{100} \; Q_1^{\,2}}{d^4}$$

SKBS calculates the normalized ΔP for a flow of 1 gpm and a fluid density at 100°F. These energy loss terms are calculated in a similar manner to the SKSB terms for placement into column 11 of BLEG array

$$\text{where } \Sigma K_{11} = \underset{\text{Expansion}}{\Delta P_{100}} + \underset{\text{Contraction}}{\Delta P_{100}}$$

The normalized ΔP is corrected for flow and density in TTL as follows:

$$\underset{\text{energy loss}}{\Delta P_{\text{corrected}}} = \underset{\text{contraction}}{\Delta P_{100}} \times \frac{\rho}{\rho_{100}} \times Q^2$$

5.2.1.4 Approximations

The turbulent friction factor is approximated by $\frac{.316}{Re^{.25}}$ which is the Blasius law.

5.2.1.5 Limitations

Not applicable.

5.2.1.6 Function - SFRICL - SFRICT - SKSB - SKBS Variable Names

Variables	Description	Dimensions
ALEN	Length of Passage or Section	In
CK	Discontinuity Factor	--
DIA	Internal Passage Diameter	In
D100	Fluid Weight Density at 100°F	Lb/Ft^3
V100	Fluid Viscosity at 100°F	Centipoise
DOLD	Internal Diameter of Previous Element	In
DELM	Internal Diameter of Element	In

5.2.1.6 (Continued)

Variables	Description	Dimensions
FRIC	Normalized Laminar Pressure Drop	PSI/GPM
FRICT	Normalized Turbulent Pressure Drop	$PSI/GPM^{1.75}$
SFRICL	Normalized Laminar Function for Frictional Resistance	PSI/GPM
SFRICT	Normalized Turbulent Function for Frictional Resistance	$PSI/GPM^{1.75}$
SKBS	Normalized Discontinuity Function from Big to Small Diameter	PSI/GPM^{2}
SKSB	Normalized Discontinuity Function from Small to Big Diameter	PSI/GPM^{2}

5.2.1.7 SFRICL - SFRICT - SKSB - SKBS Function Listing

```
      FUNCTION SFRICL(V100,ALEN,D100,DIA)
C             LAMINAR REFERENCE PRESSURE DROP     DATE 9/03/74
      FRIC=3.64E-7*V100*ALEN*D100*(DIA**(-4))
      SFRICL=FRIC
      RETURN
      END
      FUNCTION SFRICT(V100,ALEN,D100,DIA)
C         TURBULENT REFERENCE PRESSURE DROP     DATE 9/03/74
      FRICT=7.591E-7*(V100**(.25))*ALEN*D100*(DIA**(-4.75))
      SFRICT=FRICT
      RETURN
      END
      FUNCTION SKSB(DOLD,DELM,D100)
C        DISCONTINUITY CALCULATION SMALL TO BIG     DATE 9/03/74
      CK=(1.-((DOLD/DELM)**2))**2
      SKSB=CK*1.799E-5*D100*(DOLD**(-4))
      RETURN
      END
      FUNCTION SKBS(DOLD,DELM,D100)
C         DISCONTINUITY CALCULATION BIG TO SMALL     DATE 9/03/74
      CK=.5*(1.-((DELM/DOLD)**2))
      SKBS=CK*1.799E-5*D100*(DELM**(-4))
      RETURN
      END
```

SECTION VI

CALCULATION AND ELEMENT SUBROUTINES

The solution procedure is totally controlled by the CALC subroutine. The subroutines used for computations are divided into two sections. Section 6.1 describes the calculation subroutines which include CALC and TTL. These subroutines comprise the solution procedure. Section 6.2 discusses the dynamic element subroutines which supply information to the calculation section.

6.1 CALCULATION SUBROUTINES

The calculation subroutines are CALC and TTL. CALC controls the entire solution procedure, and it computes the pressures and flows for the pressure, return and suction systems of the modeled system. TTL updates all the dynamic elements in the entire system and computes pressure drops in legs using the current flow guess. Figure 36 is a general solution procedure flow chart for the calculation subroutines.

The CALC subroutine is called from the Main Program SSFAN. The iteration counters, flow tolerance, and computational arrays are initialized. The actual calculation phase begins with a call to the TTL subroutine with the initial flow guesses in the legs. The legs conductances are computed and a call is made to SIMULT. New flows are calculated from the pressures and compared with the old. If the convergence criteria is not met, a new iteration is started.

After the system has converged one more iteration is performed because some dynamic element calculations use the previous iteration pressure or flow values.

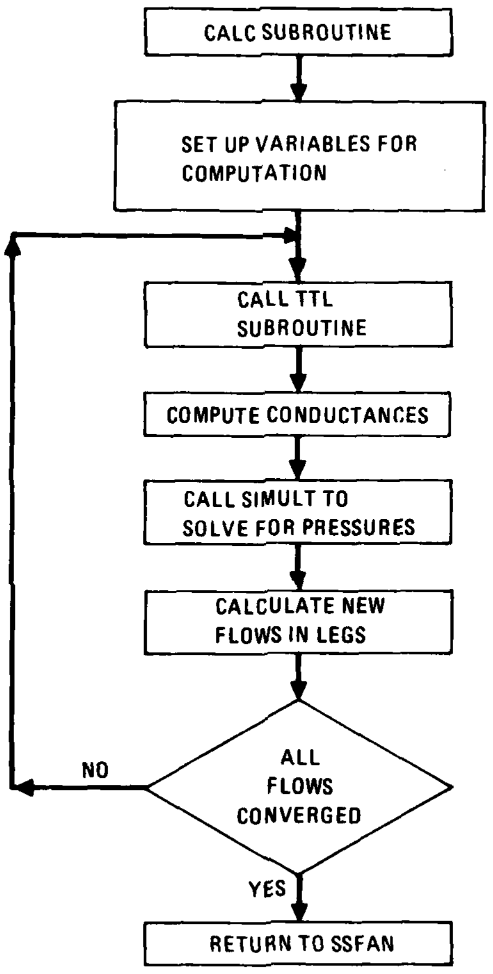

CALC SUBROUTINE
SET UP VARIABLES FOR COMPUTATION
CALL TTL SUBROUTINE
COMPUTE CONDUCTANCES
CALL SIMULT TO SOLVE FOR PRESSURES
CALCULATE NEW FLOWS IN LEGS
ALL FLOWS CONVERGED
NO
YES
RETURN TO SSFAN

FIGURE 36
GENERAL SOLUTION PROCEDURE FLOW CHART FOR THE CALCULATION SUBROUTINES

GP03-0594-1

6.1.1 CALC SUBROUTINE

The CALC subroutine is responsible for the steady state calculations in the system. CALC is called from the SSFAN Main Program - SSFAN. The subroutine computes the pressures at all the system pressure points and flows in all the legs, using resistance coefficients obtained from the TTL subroutine. Figure 37 is a generalized flow diagram of CALC.

On entry into CALC the CALC1 and CALC2 storage location identification arrays are built, the CALC1 and CALC2 arrays are zeroed, then the computation phase begins. All the dynamic elements have resistance coefficients calculated from the TTL subroutine and stored in BLEG array, then the leg conductances are calculated. These conductance values, along with constant factors, are then inserted into CALC1 and CALC2 arrays. The SIMULT subroutine is called to compute the new pressure values using the compressed matrix technique. These pressure values at the pressure points are used to calculate the new flow rates for the legs in system. When all the flows pass the convergence test, program control is passed to SSFAN. Should the number of iterations exceed 60, CALC terminates with the most recent values of flow and pressure and prints out the iteration count.

6.1.1.1 Math Model

A step-by-step procedure of the solution process is outlined below for the CALC subroutine. A detailed mathematical explanation is found in Appendix A of this manual.

The CALC subroutine controls the entire solution procedure. The process can be summarized in seven steps:

1. Build the CALC1 and CALC2 storage location identification arrays.

2. Zero the CALC1 and CALC2 arrays.

3. Call the dynamic element subroutines and compute resistance coefficients for laminar, turbulent or transition flow through the TTL subroutine.

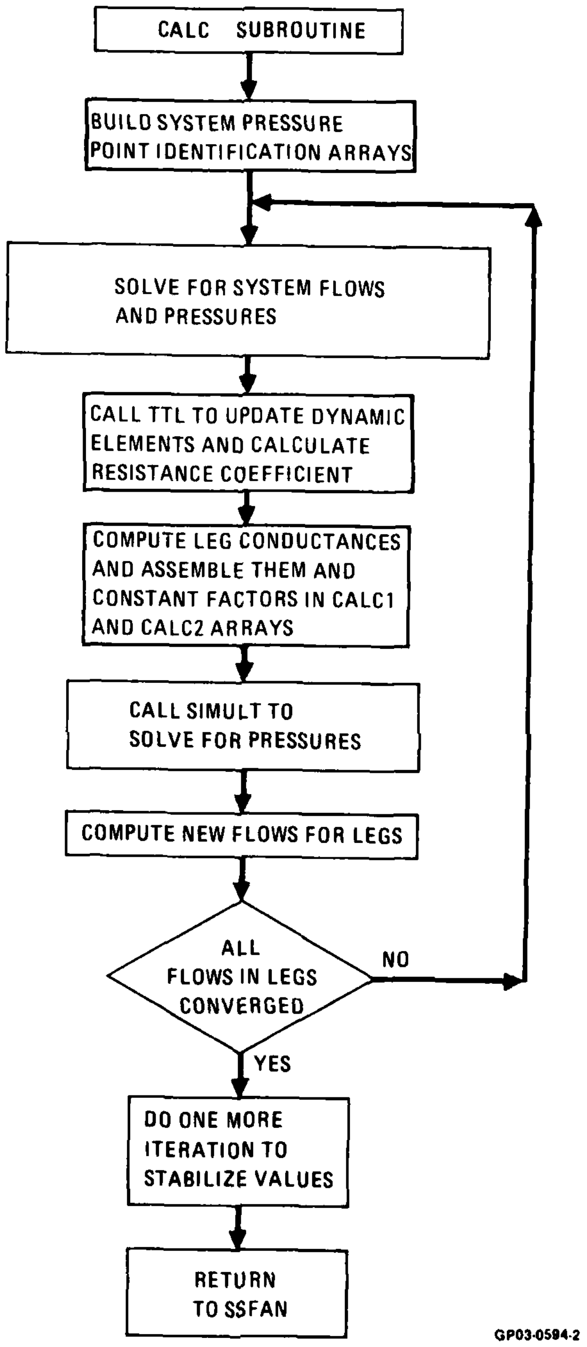

CALC SUBROUTINE
BUILD SYSTEM PRESSURE POINT IDENTIFICATION ARRAYS
SOLVE FOR SYSTEM FLOWS AND PRESSURES
CALL TTL TO UPDATE DYNAMIC ELEMENTS AND CALCULATE RESISTANCE COEFFICIENT
COMPUTE LEG CONDUCTANCES AND ASSEMBLE THEM AND CONSTANT FACTORS IN CALC1 AND CALC2 ARRAYS
CALL SIMULT TO SOLVE FOR PRESSURES
COMPUTE NEW FLOWS FOR LEGS
ALL FLOWS IN LEGS CONVERGED
NO
YES
DO ONE MORE ITERATION TO STABILIZE VALUES
RETURN TO SSFAN
GP03-0594-2

FIGURE 37
CALC GENERALIZED FLOW DIAGRAM

4. Calculate the leg conductances and input them and other constant factors from the dynamic element subroutines in the CALC1 and CALC2 arrays.

5. Call SIMULT using the compressed matrix technique to solve the system of linear algebraic equations for the variable pressure points.

6. Using the newly calculated pressures from the matrix solution compute new flow rates for the legs.

7. Compare the new flows with the old. If the convergence criteria is satisfied, or the solution procedure exceeded 60 iterations, return to the Main Program. Otherwise go back to Step 1.

A computation made in the solution of the steady state values in CALC is the calculation of QNEW. The purpose of this is to establish an error tolerance in flows that is reduced through iterations to meet the convergence criteria as discussed in Section 3.2. The majority of the CALC subroutine handles the bookkeeping necessary to manipulate the leg and pressure point numbers in the system to a version that is easier for computations by CALC.

6.1.1.2 Assumptions - See Appendix A.

6.1.1.3 Computations

On entering the CALC subroutine, the iteration counters and flow tolerance is set. Then the system pressure point identification arrays are built. A detailed development of the system pressure point identification arrays is found in Appendix A of this manual.

In the computation phase the CALC subroutine begins with initializing the conductance matrix - CALC1, and the constant matrix - CALC2, to zero values. (See Figure 38 for a flow diagram of the computation phase operation.) A call is now made to the TTL subroutine for each iteration.

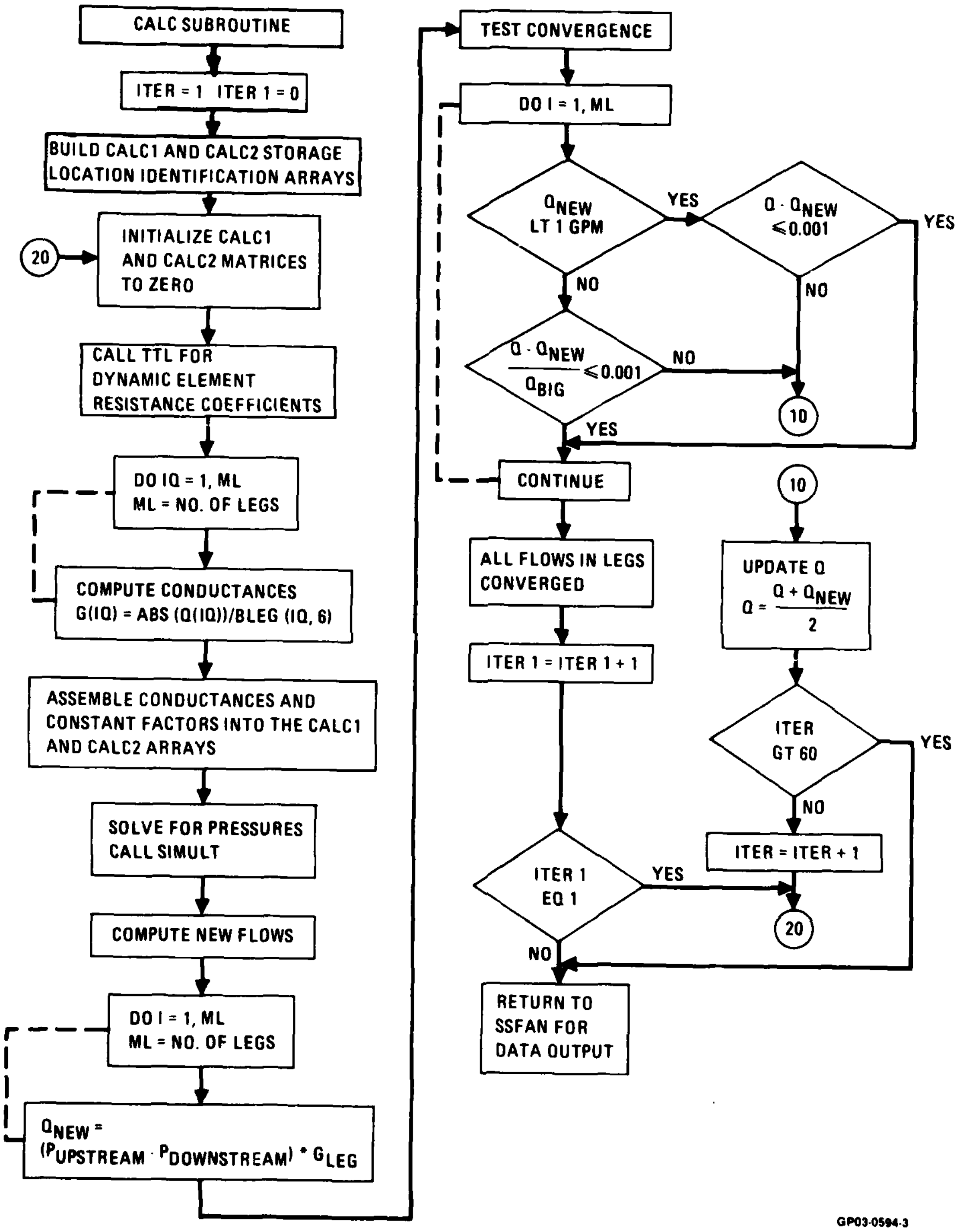

CALC SUBROUTINE
ITER = 1 ITER 1 = 0
BUILD CALC1 AND CALC2 STORAGE LOCATION IDENTIFICATION ARRAYS
20
INITIALIZE CALC1 AND CALC2 MATRICES TO ZERO
CALL TTL FOR DYNAMIC ELEMENT RESISTANCE COEFFICIENTS
DO IQ = 1, ML ML = NO. OF LEGS
COMPUTE CONDUCTANCES G(IQ) = ABS (Q(IQ))/BLEG (IQ, 6)
ASSEMBLE CONDUCTANCES AND CONSTANT FACTORS INTO THE CALC1 AND CALC2 ARRAYS
SOLVE FOR PRESSURES CALL SIMULT
COMPUTE NEW FLOWS
DO I = 1, ML ML = NO. OF LEGS
Q_NEW = (P_UPSTREAM - P_DOWNSTREAM) * G_LEG
TEST CONVERGENCE
DO I = 1, ML
Q_NEW LT 1 GPM
YES
Q - Q_NEW ≤0.001
YES
NO
NO
(Q - Q_NEW)/Q_BIG ≤0.001
NO
10
YES
CONTINUE
10
ALL FLOWS IN LEGS CONVERGED
UPDATE Q Q = (Q + Q_NEW)/2
ITER 1 = ITER 1 + 1
ITER GT 60
YES
NO
ITER = ITER + 1
ITER 1 EQ 1
YES
20
NO
RETURN TO SSFAN FOR DATA OUTPUT
GP03-0594-3

FIGURE 38
CALC SUBROUTINE OPERATION

The first and last leg numbers of the system to be solved by CALC are passed through the subroutine arguments. TTL returns the value of leg pressure drop to the BLEG array in column six.

The conductance values are calculated for each leg.

G(IQ) = ABS(Q(IQ))/BLEG(IQ,6) (1)

where IQ = leg number

ABS(Q(IQ)) = Absolute value of flow in leg IQ

BLEG(IQ,6) = Pressure drop in leg IQ

G(IQ) = Conductance of leg IQ

The conductance values must now be entered into the CALC1 matrix. Each variable pressure point is worked on individually. The main diagonal element of CALC1 contains the sum of all the leg conductances of pressure point J. Each off diagonal element equals the sum of all conductances around the pressure point J for multiple legs connected to common pressure points, and the sum of conductances for each pressure point connected to pressure point J.

The CALC2 matrix contains the constant terms of the system of linear equations that describe the model. Constant pressure drops in legs, external flows and constant pressure sources are all inserted into this matrix. Any constant pressure source or pressure drop is multiplied by the conductance of the leg it is associated with. If leg (5) has a pressure drop term - BLEG(,5), then the BLEG(,5) is multiplied by the conductance for leg (5) which is G(5), making the resulting term a flow. Thus, all external flows have no multiplication factors.

With both CALC1 and CALC2 filled, the SIMULT subroutine is called to solve for pressures in the system. The answers come back in the first column of the CALC1 matrix and are put into PQL array, column 1, which contains all the system pressures. A new flow is calculated for each leg in the system based on the recent calculation of the pressures. The new flow is equal to the difference of pressures on the end points of the leg plus any constant pressure drops times the conductance of the leg.

The solution for flows in all the legs are final when all the previous flows (Q) and the latest calculated flows (QNEW) are within a specified tolerance. If both the old and new flows are less than one GPM then

$$\text{ABS}\ (\text{Q} - \text{QNEW}) \leq .001 \qquad (2)$$

is the tolerance. For flows greater than or equal to one GPM,

$$\text{ABS}\ \frac{\text{Q} - \text{QNEW}}{\text{QBIG}} \leq .001 \qquad (3)$$

where QBIG equals the larger of Q or QNEW, is the convergence criteria. If equations (2) or (3) are not satisfied in each leg of the system a new value of flow will be compared in each leg by the following equation:

$$\text{Q}' = \frac{\text{Q} + \text{QNEW}}{2} \qquad (4)$$

These new flows are given to TTL for computation of new dynamic element resistance values for another iteration. If all the legs do not converge after sixty iterations, the cycle stops and all the current values are used as the steady-state variables. Before transfer is made back to SSFAN a last iteration is made to stabilize pressure drops and flows for the steady state conditions.

6.1.1.4 Approximations

The coefficients of the CALC1 matrix are linearily approximated to represent the system conductances. Inherent approximations exist in some of the constant data in CALC2.

6.1.1.5 Limitations

Most limitations exist in the areas of physical discontinuities. CALC was written to solve a flow balance in a system. Any flow discontinuities that occur, such as in a simple unbalanced actuator, must have mathematical formula to describe what happens to the flow. CALC also requires the leg pressure drops to be continuous over a specified flow range. When this does not occur, as in a check valve, the proper input from the check valve subroutine must be fed to CALC so it may respond to the changed conditions. Please refer to Appendix A for a more thorough discussion on the limitations of CALC.

6.1.1.6 CALC Variable Names

Variables	Description	Dimensions
ALT	Attitude	FT
BLEG	General purpose array	--
BRANCHP	Array containing dynamic element data	--
CALC1	MxM array of conductances	--
CALC2	M array of constants	--
CJCT	Pump case drain port junction number	--
D	Dummy storage array of JCOL values	--
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
D100	Weight density at 100°F	LB/FT^3
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
G	Array of conductances	--
I, IC	Integer counters	--
ICENT	Array containing number of non-zero elements in each row	--
ICOL	Array containing the column number of each non-zero element	--
IDIAG	Array which identifies which columns of CALC1 contain positive conductance values	--
IERROR	Error indicator if $\neq 0$	--
IJ, IM	Integer counters	--
ILEP	Array of leg numbers with the corresponding pressure on each end	--
INEG	Array which stores the second appearance of a negative conductance value	--
IORDER	Array giving pivot selection based on min-row min-col criteria	--
IQ, IT	Integer counters	--
IRENT	Array containing number of non-zero elements in each row	--

6.1.1.6 (Continued)

Variables	Description	Dimensions
ITER	Iteration counter	--
ITER1	Final iteration counter	--
IU,IV,IZ,I1,J,J1, J2,J3	Integer counters	--
JCENT	Array which identifies the number of non-zero entries in each column of CALC1	--
JCOL	Final array which identifies the columns in a square CALC1 array which are filled with non-zero terms	--
JCT	Pump case drain port junction number	--
JEM	Number of pressure points	--
JNEG	Array which identifies which column in CALC1 contains the first appearance of a negative conductance value in CALC1 array	
JRENT	Array which identifies the number of non-zero entries in each row of CALC1	--
KOUNT	Integer Variable	--
K,K1,K2,K3,K4, K5,K6,K8,K9	Integer counters	--
K7	Temporary storage of non-zero entries in each row of CALC1	--
L,LM,L1	Integer counters	--
MAXITER	Maximum number of iterations	--
ML	Total number of legs	--
N	Branch point number of actuator	--
N1	Integer counters	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG and ILEP arrays	--
NPQ	Total number of rows in POL array	--
NPQL2	Array containing corresponding pressure point numbers for constant pressure values of array POL2	--

6.1.1.6 (Continued)

Variables	Description	Dimensions
PAMB	Atmospheric ambient pressure	PSI
PQL	Array of pressure points, flows and port numbers	--
PQL2	Array containing constant pressure values	--
QBIG	Larger of Q or QNEW	--
QNEW	Latest values of leg flows	--
TEMP	Fluid temperature	°F
TEST	Largest positive value in D( ) array	--
TEST2	Location number in D( ) array containing largest positive value	--
TOL	Tolerance for flow balancing	--
TY	Element type	--
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
V100	Fluid viscosity at 100°F	CENTISTOKES

6.1.1.7 CALC Subroutine Listing

```
      SUBROUTINE CALC(ML,N,JEM,BLEG,PQL,CALC1,JCOL,CALC2,JRENT,JCENT,
     1IDIAG,JNEG,INEG,BRANCHP,NBP2,NL,ILEP,IRENT,ICENT,IORDER,ICOL,
     2NPQ)
C                    CALC                            1 DEC 79
C      CALC COMPUTES PRESSURES AND FLOWS IN ALL SYSTEMS
       COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
       COMMON /BLK3/NPQL2(20),PQL2(20)
       COMMON /BLK7/IERROR,ITER
       DIMENSION D(5)
       DIMENSION CALC1(1),CALC2(1),JCOL(1),JRENT(1),JCENT(1),ICOL(1)
       DIMENSION IDIAG(1),JNEG(1),INEG(1),BRANCHP(1)
       DIMENSION ICENT(1),IRENT(1),IORDER(1)
       DIMENSION BLEG(1),ILEP(1),PQL(1)
       MAXITER=150
       TOL=.001
     1 ITER=1
       IC=0
       ITER1=0
       MARK=0
       IM=0
       DO 2 I=1,ML
       IF(BLEG(I+(2*NL)).LT.1.)BLEG(I+(2*NL))=1.
       DO 2 J=1,NBP2
       TY=BRANCHP(J)
       IF(TY.NE.5.)GO TO 2
       JCT=BRANCHP(J+3*NBP2)
       IF(JCT.EQ.BLEG(I+NL))BLEG(I+2*NL)=(-1.)
     2 CONTINUE
C      PRINT,'BLEG'
C      DO 888 I=1,ML
C888   WRITE(108,999)(BLEG(I+(J-1)*NL),J=1,16)
C999   FORMAT(16F8.3)
C      STORE NON-ZERO COLUMN NUMBERS IN JCOL
       DO 3 K=1,ML
       I=ILEP(K)
       J=ILEP(K+NL)
       DO 3 K1=1,2
       J1=I
       IF (K1.EQ.2)J1=J
       DO 3 K2=1,2
       J2=I
       IF (K2.EQ.2)J2=J
       DO 3 K3=1 5
       J3=JCOL(J1+(K3-1)*NL)
       IF (J3.NE.0)GO TO 3
       JCOL(J1+(K3-1)*NL)=J2
       J2=0
     3 IF (J3.EQ.J2)J2=0
C      LOAD CONSTANT PRESSURE NODES INTO JCOL
       N1=N
       IF(NPQL2(N+1).NE.0)N1=N+1
     4 DO 6 K=1,N1
       I1=NPQL2(K)
       DO 5 J1=1,5
```

6.1.1.7 (Continued)

```
    5 JCOL(I1+(J1-1)*NL)=0
    6 JCOL(I1)=I1
C     BUILD JRENT,JCENT; REORDER JCOL; BUILD IDIAG
      DO 12 K=1,JEM
      KOUNT=0
      DO 7 K8=1,5
      DO 7 K9=1,JEM
    7 IF (JCOL(K9+(K8-1)*NL).EQ.K)KOUNT=KOUNT+1
      JCENT(K)=KOUNT
      KOUNT=0
      DO 8 K1=1,5
    8 IF (JCOL(K+(K1-1)*NL).NE.0)KOUNT=KOUNT+1
      JRENT(K)=KOUNT
      DO 9 K2=1,KOUNT
    9 D(K2)=JCOL(K+(K2-1)*NL)
      DO 11 K4=1,KOUNT
      TEST=0
      DO 10 K5=1,KOUNT
      IF (D(K5).LT.TEST)GO TO 10
      TEST=D(K5)
      TEST2=K5
   10 CONTINUE
      K6=KOUNT+1-K4
      JCOL(K+(K6-1)*NL)=TEST
   11 D(TEST2)=0
      K7=JRENT(K)
      DO 12 KOUNT=1,K7
   12 IF (JCOL(K+(KOUNT-1)*NL).EQ.K)IDIAG(K)=KOUNT
C     BUILD INEG,JNEG
      DO 14 K=1,ML
      I=ILEP(K)
      J=ILEP(K+NL)
      I1=JRENT(I)
      J1=JRENT(J)
      INEG(K)=0
      JNEG(K)=0
      DO 13 KOUNT=1,I1
   13 IF (JCOL(I+(KOUNT-1)*NL).EQ.J)JNEG(K)=KOUNT
      DO 14 KOUNT=1,J1
   14 IF (JCOL(J+(KOUNT-1)*NL).EQ.I)INEG(K)=KOUNT
   15 CONTINUE
C     CALL OPUT2(ILEP,BLEG,NL,PQL,NPQ)
C     INITIALIZE CALC1 AND CALC2 ARRAYS TO ZERO
      DO 16 K=1,ML
      BLEG(K+4*NL)=0.
   16 BLEG(K+11*NL)=0.
      DO 17 K=1,JEM
   17 PQL(K+NBP2)=0.
      DO 19 L1=1,JEM
      DO 18 K1=1,9
   18 CALC1(L1+(K1-1)*NL)=0.
   19 CALC2(L1)=0.
C     COMPUTE CONDUCTANCES FOR CALC ARRAYS
```

6.1.1.7 (Continued)

```
      CALL TTL(ML,NL,BLEG,BRANCHP,NBP2,ILEP,PQL,NPQ)
C     DO 887 I=1,ML
C887  WRITE(6,999)(BLEG(I+(J-1)*NL),J=1,16)
      IF(IERROR.GT.0)RETURN
      DO 20 IQ=1,ML
      G=BLEG(IQ+5*NL)
      Q=ABS(BLEG(IQ+2*NL))
   20 BLEG(IQ+5*NL)=Q/G
C     BUILD THE CALC1 ARRAY
      DO 21 K=1,ML
      I=ILEP(K)
      J=ILEP(K+NL)
      L=IDIAG(I)
      LM=IDIAG(J)
      CALC1(I+(L-1)*NL)=CALC1(I+(L-1)*NL)+BLEG(K+5*NL)
      CALC1(J+(LM-1)*NL)=CALC1(J+(LM-1)*NL)+BLEG(K+5*NL)
      L=JNEG(K)
      LM=INEG(K)
      IF (L.NE.0)CALC1(I+(L-1)*NL)=CALC1(I+(L-1)*NL)-BLEG(K+5*NL)
   21 IF (LM.NE.0)CALC1(J+(LM-1)*NL)=CALC1(J+(LM-1)*NL)-BLEG(K+5*NL)
C     BUILD CALC2 ARRAY
      N1=N
      IF(NPQL2(N+1).NE.0)N1=N+1
      DO 22 I=1,N
      I1=NPQL2(I)
      CALC1(I1)=1.
   22 CALC2(I1)=PQL(I1)-PQL(I1+NPQ)
      DO 23 I=1,JEM
   23 CALC2(I)=CALC2(I)+PQL(I+NPQ)
      DO 24 I=1,ML
      IF(BLEG(I+4*NL).EQ.0.)GO TO 24
      CALC2(ILEP(I))=CALC2(ILEP(I))-BLEG(I+4*NL)*BLEG(I+5*NL)
      CALC2(ILEP(I+NL))=CALC2(ILEP(I+NL))+BLEG(I+4*NL)*BLEG(I+5*NL)
   24 CONTINUE
      IF(NPQL2(N+1).EQ.0)GO TO 25
      J=NPQL2(N+1)
      CALC2(J)=.001
   25 CALL SIMULT(JEM,ITER,CALC1,CALC2,JCOL,JRENT,JCENT,ICENT,
     1IRENT,IORDER,ICOL,NPQ)
      DO 26 IM=1,JEM
      PQL(IM)=CALC1(IM)
   26 CONTINUE
C     CALCULATE NEW FLOW RATES
      DO 27 IT=1,ML
      IU=ILEP(IT)
      IV=ILEP(IT+NL)
      BLEG(IT+6*NL)=(PQL(IU)+BLEG(IT+4*NL)-PQL(IV))*BLEG(IT+5*NL)
   27 CONTINUE
C     TEST NEW FLOW RATES
      DO 31 IJ=1,ML
      Q=BLEG(IJ+2*NL)
      QNEW=BLEG(IJ+6*NL)
      IF(ABS(Q-QNEW).LE.TOL)GO TO 31
      IF(ABS(Q).GT.1.)GO TO 28
      IF(ABS(QNEW).LE.1.)GO TO 35
```

6.1.1.7 (Continued)

```
   28 IF(ABS(Q).GT.ABS(QNEW))GO TO 29
      QBIG=QNEW
      GO TO 30
   29 QBIG=Q
   30 IF(ABS((Q-QNEW)/QBIG).GT.TOL)GO TO 35
   31 CONTINUE
      IF(ITER1.GE.1)GO TO 32
      ITER1=ITER1+1
      GO TO 35
   32 CONTINUE
      DO 33 IZ=1,ML
      Q=BLEG(IZ+2*NL)
      IF(Q.EQ.0.)BLEG(IZ+2*NL)=.00001
   33 CONTINUE
      IF(ITER.GE.MAXITER)WRITE(6,34)
      IF(ITER.GE.MAXITER)STOP
   34 FORMAT(10X,42HEXCEEDED MAX NO OF ITERATIONS-CHECK SYSTEM)
      IC=0
      CALL FLOCHEK(IC,BRANCHP,NBP2,BLEG,NL,PQL)
      CALL VCHEK(ML,IC,BRANCHP,NBP2,BLEG,NL,ILEP,PQL,NPQ)
      CALL ACTCHEK(N,IC,BRANCHP,NBP2,PQL,NPQ)
      CALL HTRCHK(BRANCHP,NBP2,IC)
      IF(IC.NE.0)GO TO 1
      RETURN
C     RECALCULATE FLOW RATES
   35 DO 36 I=1,ML
      Q=BLEG(I+2*NL)
      QNEW=BLEG(I+6*NL)
      BLEG(I+2*NL)=(Q+QNEW)/2.
      IF(Q.EQ.0.)BLEG(I+2*NL)=.00001
   36 CONTINUE
      IF(ITER.EQ.MAXITER)GO TO 32
      ITER=ITER+1
      GO TO 15
      END
```

6.1.2 TTL Subroutine

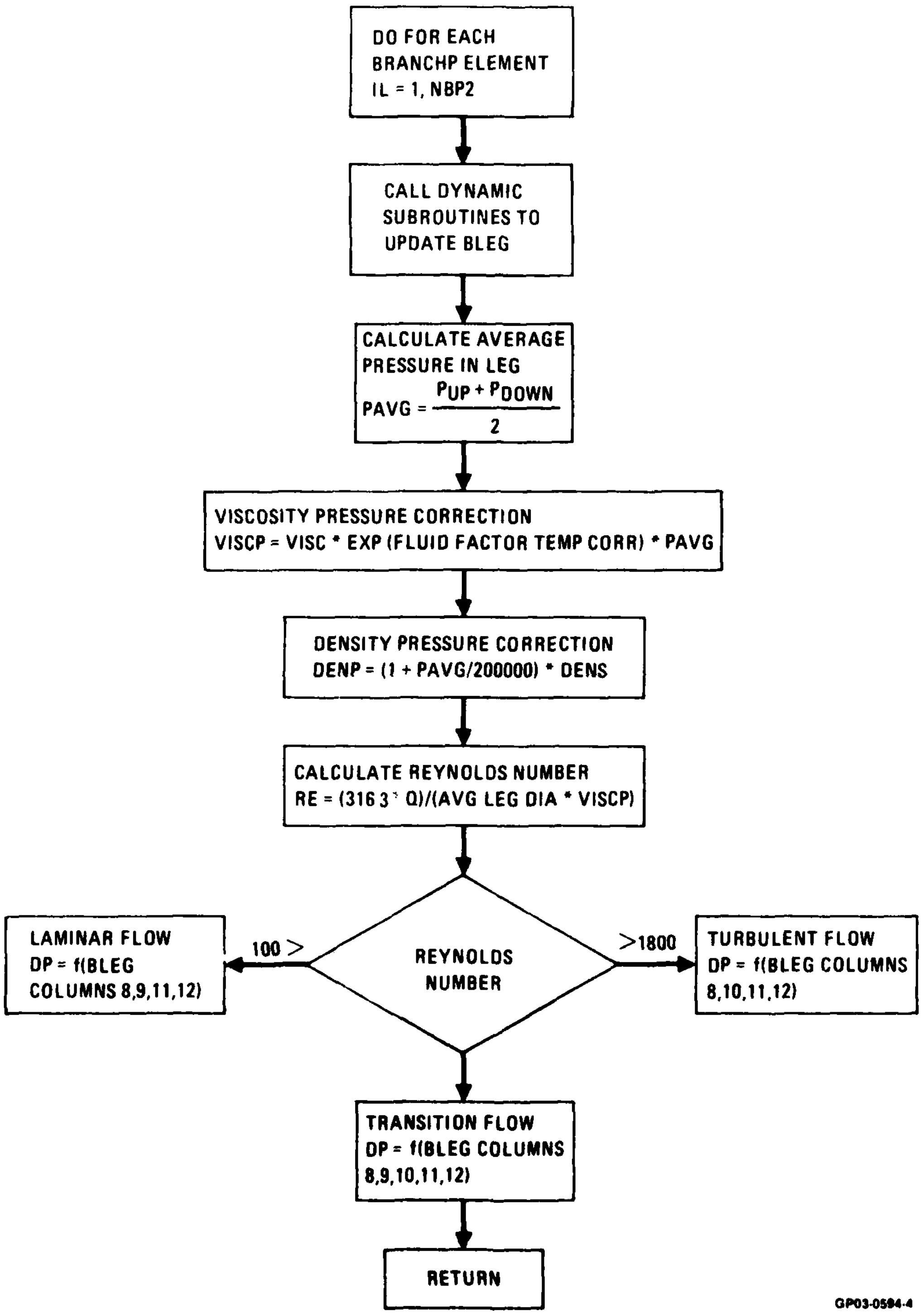
The TTL subroutine updates all the system leg values by calling the dynamic element subroutines. The dynamic subroutines give pressure at a pressure point, flow in a leg, or the change in pressure (ΔP) in a leg. These values returned by the element subroutines are found in column 5 or column 12 of the BLEG array or columns 1 or 2 of the PQL array. This value in column 5 of BLEG represents the ΔP term. The one in column 12 corresponds to energy loss coefficients in a leg due to valves and orifices. PQL column 1 is the pressure point value and PQL column 2 is a flow loss or gain at a pressure point.

Once the dynamic elements in a leg are called, viscosity and density corrections factors, and the leg Reynolds number are calculated using the average leg pressure. The Reynolds number determines the proper equation to be used to account for all the element pressure drops as a function of leg flow. One of three equations are generated depending on whether the flow is laminar, turbulent, or in transition. The resulting equation is evaluated at the leg flow rate and the pressure drop value is set into the sixth column of the BLEG array.

Each leg in the system is processed in the same manner. First, the leg's dynamic elements are updated then the leg pressure drop is calculated and stored in BLEG. After the final leg, program control is passed back to the calling subroutine. A descriptional flow chart of the TTL subroutine is shown in Figure 39.

6.1.2.1 Math Model

The average pressure for a leg is calculated by using the upstream and downstream pressures found in the PQL array. The location of the pressure


DO FOR EACH BRANCHP ELEMENT IL = 1, NBP2
CALL DYNAMIC SUBROUTINES TO UPDATE BLEG
CALCULATE AVERAGE PRESSURE IN LEG
PAVG = (PUP + PDOWN) / 2
VISCOSITY PRESSURE CORRECTION
VISCP = VISC * EXP (FLUID FACTOR TEMP CORR) * PAVG
DENSITY PRESSURE CORRECTION
DENP = (1 + PAVG/200000) * DENS
CALCULATE REYNOLDS NUMBER
RE = (3163 * Q)/(AVG LEG DIA * VISCP)
REYNOLDS NUMBER
100 >
>1800
LAMINAR FLOW DP = f(BLEG COLUMNS 8,9,11,12)
TURBULENT FLOW DP = f(BLEG COLUMNS 8,10,11,12)
TRANSITION FLOW DP = f(BLEG COLUMNS 8,9,10,11,12)
RETURN

GP03-0594-4

FIGURE 39
TTL SUBROUTINE DESCRIPTIONAL FLOW CHART

points in PQL for each leg is found in the ILEP array.

PAVG = (PQL(ILEP(IQ,2),1)+PQL(ILEP(IQ,3),1))/2. (1)

where IQ = the current leg number

The average pressure is used in the calculation of a viscosity-pressure correction factor VISCP.

VISCP=VISC*EXP(FLUIDK*PAVG) (2)

where VISC = Fluid Viscosity at System Operating Temperature and Atmospheric Pressure

FLUIDK = Fluid Correction Factor - Temperature Compensated

PAVG is also used for a density-pressure correction factor DENP.

DENP = (1. +(PAVG/200000.))*DENS (3)

where DENS = Fluid Density at System Temperature and Atmospheric Pressure

Since the leg resistance coefficients are adjusted to a reference viscosity and density at a temperature of 100°F, the VISCP and DENP correction terms are

CFVP = VISCP/V100 (4)

and

CFDP = DENS/D100 (5)

The Reynolds number is calculated for each LEG by the following equation:

RE = (3163.*F)/DIA*VISCP (6)

where F = Absolute value of flow in a leg (GPM)

DIA = Leg average diameter (IN)

The calculation of the Reynolds number is used to determine the LEG flow equation. Pressure drop (DP) for each leg is calculated using columns 8 thru 12 of the BLEG array. The DP equation is of the form:

$$DP = K_1 + K_2Q + K_3Q^{1.75} + K_4Q^2 + K_5Q \qquad (7)$$

where

K_1	=	BLEG(IQ,8) fixed constant pressure drops
K_2	=	BLEG(IQ,9) laminar resistance coefficients
K_3	=	BLEG(IQ,10) turbulent resistance coefficients
K_4	=	BLEG(IQ,11) local energy loss coefficients
K_5	=	BLEG(IQ,12) leakage resistance coefficients
IQ	=	Leg Number

The K_2 and K_3 terms are corrected with both the viscosity-pressure coefficient (CFVP) and the density-pressure coefficient (CFDP). K_4 is only density-pressure corrected and K_5 is viscosity-pressure corrected.

Laminar flows for Reynolds numbers less than 100 use columns 8, 9, 11 and 12 of BLEG for the DP calculation. Turbulent flows with Reynolds numbers greater than 1800 use BLEG columns 8, 10, 11 and 12. For transition flows between 100 and 1800 a maximum laminar and minimum turbulent flow is calculated using 100 and 1800 for the Reynolds numbers in equation (6) and solving for flows. These flows yield a DPMAXL and DPMINT terms when inserted into the proper pressure drop equation. QMAXL and DPMAXL define a point on a Flow vs Pressure Drop graph. QMINT and DPMINT define the other point. The value for the leg pressure drop results from interpolating between these two points using the current leg flow rate. For a DPMINT greater than 20000 the laminar equation only is used to give the leg pressure drop.

6.1.2.2 Assumptions - None.

6.1.2.3 Computations

The DO loop parameter is NBP2, the number of dynamic elements in the system. The dynamic elements are updated for each iteration. The DO loop is incremented and another dynamic element is done in a similar manner, until all dynamic elements are processed. Next leg pressure drops are calculated. Each leg pressure drop is calculated for each iteration and passed to BLEG column 6.

6.1.2.4 Approximations - Not applicable.

6.1.2.5 Limitations - Not applicable.

6.1.2.6 TTL Variable Listing

Variable	Description	Dimension
ALT	Altitude	FT
BLEG	General purpose array	--
BRANCHP	Dynamic element data storage array	--
CFDP	Pressure corrected density coefficient	--
CFVP	Pressure corrected viscosity coefficient	--
DENP	Density-pressure correction factor	--
DENS	Fluid density at system temperature and atmospheric pressure	LB/FT^3
DIA	Leg diameter	IN^2
DP	Leg pressure drop	--
DPXL	Laminar pressure drop	PSID
DPXT	Turbulent pressure drop	PSID
D100	Density at 100°F	LB/FT^3
F	Absolute value of leg flow	GPM
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
IERROR	Error indicator if ≠0	--
IL	Element row location in BRANCHP	--
ILEP	Array of leg numbers with the up and downstream pressure points	--
IQ	Leg number	--
ITER	Iteration counter	--
ML	Total number of legs	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG and ILEP arrays	--

Variable	Description	Dimension
NPQ	Total number of rows in PQL array	--
PAMB	Atmospheric ambient pressure	PSI
PAVG	Average pressure of a leg	PSI
PCT	Percent	--
PQL	Array of pressure points, flows and port numbers	--
QMAXL	Maximum laminar flow	GPM
QMINT	Minimum turbulent flow	GPM
Q1200	Flow when Reynolds Number equals 1200	GPM
RE	Reynolds Number	--
TEMP	Fluid temperature	°F
TLAM	Reynolds Number at which flow is totally laminar	--
TTURB	Reynolds Number at which flow is totally turbulent	--
TY	Element type	--
VISC	Fluid viscosity at system operating temperature and atmospheric pressure	--
VISCP	Viscosity - pressure correction factor	--
V100	Viscosity at 100°F	CENTISTOKES

6.1.2.7 TTL Subroutine Listing

```
      SUBROUTINE TTL(ML,NL,BLEG,BRANCHP,NBP2,ILEP,PQL,NPQ)
C                            TTL                          1 DEC*79
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION ILEP(1),PQL(1),BLEG(1),BRANCHP(1)
      COMMON /BLK6/VISCP,DENP
      COMMON /BLK7/IERROR,ITER
      TLAM=100.
      TTURB=1800.
      IF(TTURB.LT.TLAM)TTURB=TLAM
      DO 1 IL=1,NBP2
      IF(BRANCHP(IL).EQ.0.)GO TO 2
      IF(BRANCHP(IL+20*NBP2).EQ.0.)GO TO 1
      TY=BRANCHP(IL)
      IF(TY.EQ.5.)CALL DPUM5(IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ,ILEP)
      IF(TY.EQ.9.)CALL DRESV(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
      IF(TY.EQ.91.)CALL DRESV(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
      IF(TY.EQ.92.)CALL DRESV(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
      IF(TY.EQ.24.)CALL DTEE24(IL,BRANCHP,NBP2,BLEG,NL)
      IF(TY.EQ.25.)CALL DCRO25(IL,BRANCHP,NBP2,BLEG,NL)
      IF(TY.EQ.4.)CALL DACT4(IL,BRANCHP,NBP2,BLEG NL,PQL,NPQ)
      IF(TY.EQ.3.)CALL DCKV3(IL,BRANCHP,NBP2,BLEG,NL)
      IF(TY.EQ.31.)CALL DRIW31(IL,BRANCHP,NBP2,BLEG,NL)
      IF(TY.EQ.33.)CALL DVRE33(IL,BRANCHP,NBP2,BLEG,NL)
      IF(TY.EQ.34.)CALL DVSO34(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
      IF(TY.EQ.35.)CALL DVSO34(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
      IF(TY.EQ.36.)CALL DVSO34(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
      IF(TY.EQ.37.)CALL DVSO34(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
      IF(TY.EQ.38.)CALL DVSO34(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
      IF(TY.EQ.10.)CALL DPEC10(IL,BRANCHP,NBP2,BLEG,NL)
      IF(TY.EQ.6.)CALL DFIL6(IL,BRANCHP,NBP2,BLEG,NL)
      IF(TY.EQ.7.)CALL DACC7(IL,BRANCHP,NBP2,PQL,NPQ)
      IF(TY.EQ.8.)CALL DMTR8(IL,BRANCHP,NBP2,PQL,NPQ,BLEG,NL)
    1 CONTINUE
    2 DO 13 IQ=1,ML
      DIA=BLEG(IQ+3*NL)
      IF(DIA.EQ.0.0)DIA=.0001
      F=ABS(BLEG(IQ+2*NL))
      PAVG=(PQL(ILEP(IQ))+PQL(ILEP(IQ+NL)))/2.
      IF(PAVG.LE.0.0)GO TO 3
      IF(PAVG.GT.20000.)PAVG=20000.
      FLUIDK=BLEG(IQ+14*NL)
      VISC=BLEG(IQ+15*NL)
      VISCP=VISC*EXP(FLUIDK*PAVG)
      GO TO 4
    3 VISCP=VISC
    4 DENP=(1.+(PAVG/200000.))*DENS
      IF(DENP.LE.0.0)DENP=DENS
      RE=(3161.77*F)/(DIA*VISCP)
      CFVP=VISCP/V100
      CFDP=DENP/D100
```

6.1.2.7 (Continued)

```
    5 DPXL=BLEG(IQ+7*NL)+F*CFVP*CFDP*BLEG(IQ+8*NL)+F*BLEG(IQ+11*NL)
      DPXL=DPXL+BLEG(IQ+10*NL)*CFDP*F**2
      DPXT=BLEG(IQ+7*NL)+BLEG(IQ+11*NL)*F
      DPXT=DPXT+CFDP*(CFVP**.25)*BLEG(IQ+9*NL)*(F**1.75)
      DPXT=DPXT+CFDP*BLEG(IQ+10*NL)*(F**2)
      IF(RE.LE.TLAM)GO TO 8
      IF(RE.GE.TTURB)GO TO 9
    6 QMAXL=(TLAM*DIA*VISCP)/3161.77
      QMINT=(TTURB*DIA*VISCP)/3161.77
      Q1200=(1200.*DIA*VISCP)/3161.77
      IF(TTURB.LE.1200.)GO TO 10
      IF(TLAM.GE.1200.)GO TO 10
      IF(RE.GT.1200.)GO TO 11
      PCT=(F-QMAXL)/(Q1200-QMAXL)
    7 DP=PCT*DPXT+(1.-PCT)*DPXL
      GO TO 12
    8 DP=DPXL
      GO TO 12
    9 DP=DPXT
      GO TO 12
   10 PCT=(F-QMAXL)/(QMINT-QMAXL)
      GO TO 7
   11 PCT=(F-Q1200)/(QMINT-Q1200)
      GO TO 7
   12 BLEG(IQ+5*NL)=DP
   13 CONTINUE
      RETURN
      END
```

6.2 DYNAMIC ELEMENT SUBROUTINES

Dynamic subroutines are called during the iteration portion of program execution. Dynamic resistance coefficients are calculated at this time because the flow direction in a leg may not be known. This difference is particularly noted in an element such as a check valve with free flow allowed in one direction and zero flow allowed in the opposite direction. Generally, a dynamic subroutine is required for all elements that have moving parts such as poppets, pistons, rotating groups, etc. During the assembly phase of the program, identifiers for pressure points and the connecting leg numbers are placed in the element data array. Also, some elementsrequire an identifier for the direction the leg was assembled through the element to determine the correct sign for flow direction. These are also generated during the assembly phase. With this information, the correct equations for calculating resistance coefficients are selected or an assumed flow condition is assigned for some elements. These resistance coefficients are dependent on flow direction, flow magnitude and pressure or pressure drop. The resistance coefficients calculated during the iteration process are recalculated each iteration and are stored in BLEG array columns 5 and 12. After the system is balanced, a check is made to see that the assumed flow condition for element types 3,4,5, 8,31,33,37 and 38 is correct. If not, the initial assumed flow condition is changed, and the system is rebalanced.

6.2.1 Subroutine DCKV3

DCKV3 is the dynamic element model for a check valve.

6.2.1.1 Math Model

The check valve is generally of the type shown below - Figure 40 with a spherical seat or conical seat. A spring is placed behind the moving poppet to ensure positive seating in the reverse flow direction. The spring determines the cracking pressure of the valve.

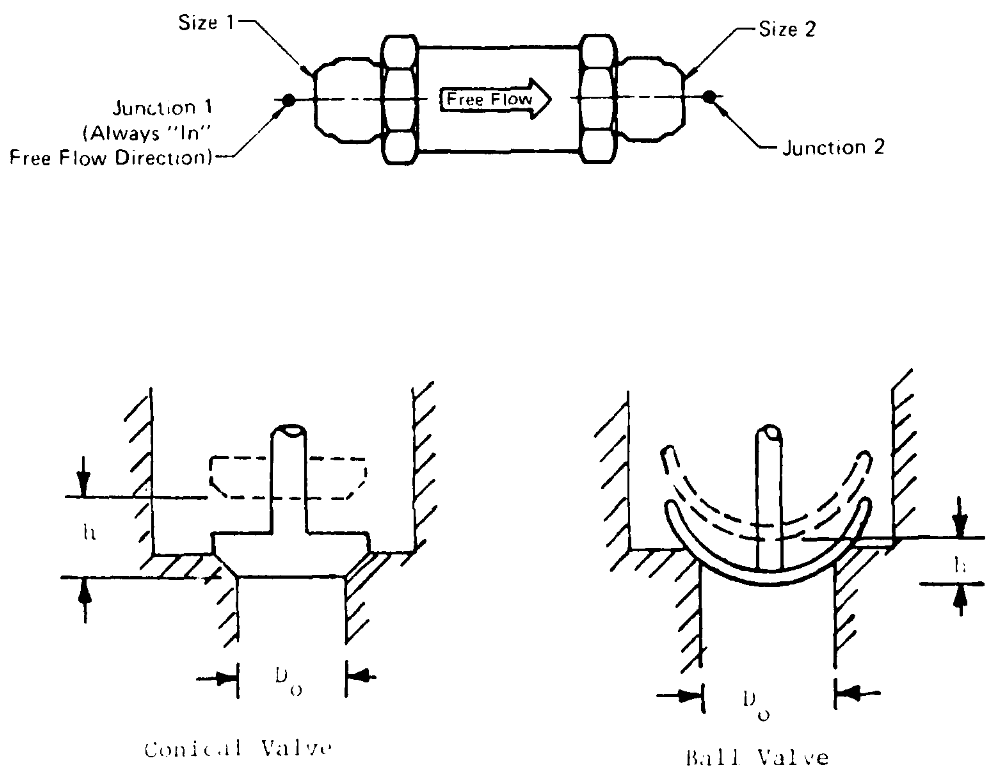


FIGURE 40
CHECK VALVE

6.2.1.2 Assumptions - Not applicable.

6.2.1.3 Computations

When DCKV3 is called from TTL subroutine, the inlet port diameter, cracking pressure and flow indicator (FI) are initialized. The flow indicator is used to establish the resistance factor to be used for calculation. On the initial flow balance, the flow is considered to be in the unchecked direction (FI=0). After the system is balanced a flow gradient check is made of the leg in which the check valve is located. If the flow gradient is opposite the initial assumed flow direction, a (-1.) is placed in the flow indicator position in BRANCHP array by VCHEK subroutine. VCHEK also passes an indicator to CALC subroutine to indicate a system rebalance is necessary. When the system is rebalanced, the high resistance factor, checked position, is used for calculation. This method has improved stability in calculation over the previous method.

If the flow is in the normal flow through direction, the equation is of the form

$\Delta P = A * (Q**1.75)$

Where A = constant

Q = flowrate

If the flow is in the checked direction, a high resistance is input as

$\Delta P = 3.E7$

The resistance coefficient is placed in BLEG column 12.

6.2.1.4 Approximations

The check valve data was derived from a manufactures flow versus pressure drop chart for various size check valves.

For sizes smaller than -4 and larger than -20, the constants for calculation are extrapolated.

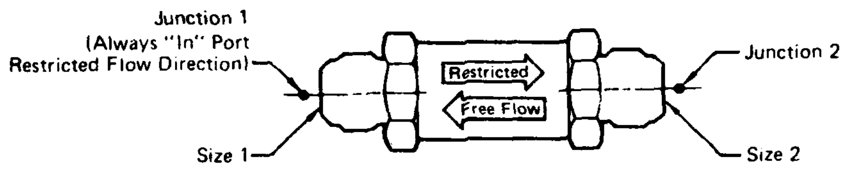
6.2.1.5 Limitations - None.

6.2.2 Entry DR1W31

DR1W31 is the dynamic element model for a one way restrictor.

6.2.2.1 Math Model

The one way restrictor schematically is similar to a check valve, Figure 41. In the free flow direction the poppet operates similar to the check valve. In the restricted flow direction the poppet is closed and flow is restricted through the orifices.



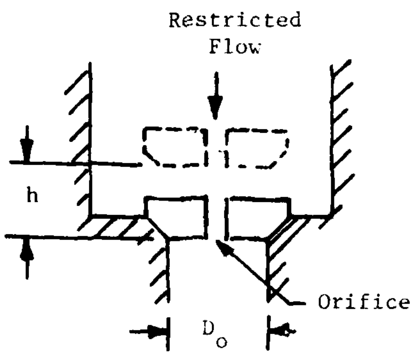


FIGURE 41
1-WAY RESTRICTOR

6.2.2.2 Assumptions - Not applicable.

6.2.2.3 Computations

The math model for the one way restrictor is similar to two other models. The free flow calculation is similar to the check valve, section 6.2.1. The restricted flow calculations are similar to the two way restrictor.

The flow is considered to be in the restricted flow direction for the initial flow balance. After the system is balanced, a check is made by VCHEK subroutine to determine whether or not the assumed flow direction was correct. If not, a (-1) is placed in the flow indicator position in BRANCHP array by VCHEK and an indicator is passed to CALC indicating a system rebalance is necessary. When the system is rebalanced the free flow direction resistance constants are used.

6.2.2.4 Approximations - Not applicable.

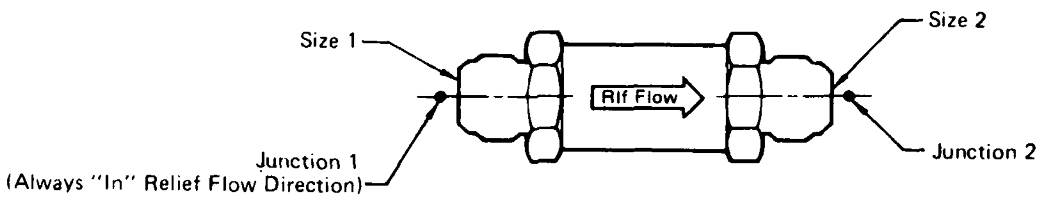
6.2.2.5 Limitations - None.

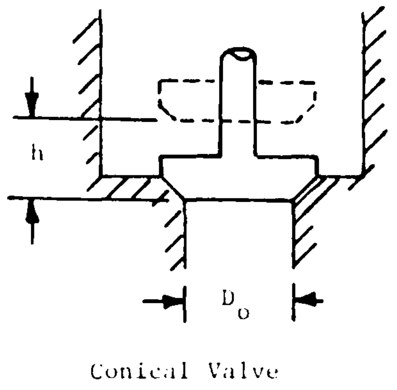
6.2.3 Entry DVRE33

DVRE33 is the dynamic element model for a relief valve.

6.2.3.1 Math Model

The relief valve is shown in Figure 42. It is similar to a check valve in operation except the spring is much stronger. The spring determines the relief valve cracking pressure.





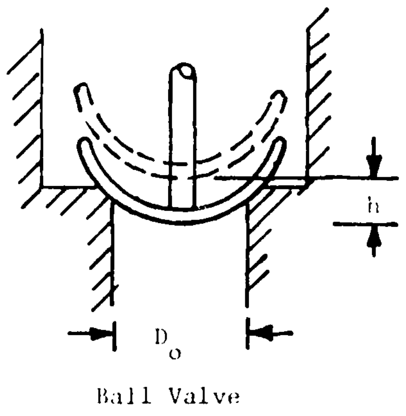


FIGURE 42
RELIEF VALVE

6.2.3.2 Assumptions - Not applicable.

6.2.3.3 Computations

When DVRE33 is called from TTL subroutine the inlet port diameter, relief pressure and flow indicator (FI) are initialized. The flow indicator is set to zero and on the initial flow balance, the relief valve will not open. After the system is balanced, VCHEK is called and a check is made to see if the assumed condition (no relief flow) was correct. If not, the flow indicator is set to (-1) in BRANCHP array and another indicator is passed to CALC subroutine to indicate a system rebalance is necessary. The calculations are similar to the check valve for relief flow conditions.

6.2.3.4 Approximations - Not applicable.

6.2.3.5 Limitations - None.

6.2.3.6 DCKV3 - DR1W31 - DVRE33 Variable Names

Variables	Description	Dimension
A	Array of constants	--
ALT	Altitude	FT
B	Check valve constants	--
BLEG	General purpose array	--
BRANCHP	Dynamic element data storage array	--
CK	Resistance coefficient	PSI/GPM
C1,C2,C3	Array input values	--
C4	Equivalent orifice diameter	IN
DENS	Fluid density at system temperature and atmospheric pressure	LB/FT^3
DP	Resistance coefficient	PSI/GPM
DP1	Resistance coefficient	PSI/GPM
D100	Viscosity at 100°F	CENTISTOKES
FI	Flow direction indicator	--
FLUIDF	Viscosity - pressure correction factor at 100°F	--
FLUIDK	Viscosity - pressure correction factor at fluid temperature	--
I	Integer counter	--
IL	Row number in BRANCHP array	--
LEG	Leg number (row in BLEG)	--
M	Row number in element array	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG array	--
PAMB	Atmospheric ambient pressure	PSI
PS	Port diameter	IN

Variables	Description	Dimension
Q	Flow rate	GPM
TEMP	Fluid temperature	°F
VISC	Fluid viscosity at system operating temperature and atmospheric pressure	--
V100	Viscosity at 100°F	CENTISTOKES

6.2.3.7 DCKV3 - DR1W31 - DVRE33 Subroutine Listing

```
      SUBROUTINE DCKV3(IL,BRANCHP,NBP2,BLEG,NL)
C        DYNAMIC RESISTANCE FOR CHK VALVE    DATE 6/13/78
      DIMENSION A(7),B(6),PS(7)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION BRANCHP(1),BLEG(1)
      DATA A/4.60873,.955169,.369931,.138549,.0549938,.0258037,.0121475/
      DATA PS/.172,.297,.391,.464,.609,.844,1.078/
      DATA B/12.,21.,27.,32.,42.,57./
      M=IL
      LEG=BRANCHP(M+20*NBP2)
      FI=BRANCHP(M+7*NBP2)
      IF(FI.LT.0.)GO TO 10
      C2=BRANCHP(M+5*NBP2)
      C1=BRANCHP(M+3*NBP2)
    1 Q=ABS(BLEG(LEG+2*NL))
      DO 2 I=1,6
      IF((C1*64.).LT.B(I))GO TO 3
    2 CONTINUE
      I=7
    3 IF(I.EQ.1)GO TO 4
      IF(I.EQ.7)GO TO 5
      DP=(A(I)-((A(I)-A(I+1))*(C1-PS(I))/(PS(I+1)-PS(I))))*Q**1.75
      GO TO 7
    4 A(I)=-29.22848*C1+9.6302
      GO TO 6
    5 A(I)=.0121475/(.03125*EXP(3.46574*(C1/1.078)))
    6 DP=A(I)*Q**1.75
    7 IF(Q.EQ.0.)BLEG(LEG+2*NL)=.0001
      IF(Q.EQ.0.)GO TO 9
      IF(DP.LT.5.)DP=5.
      DP=DP-5.+C2
      DP1=DP/Q
    8 BLEG(LEG+11*NL)=BLEG(LEG+11*NL)+DP1
      RETURN
    9 BLEG(LEG+4*NL)=BLEG(LEG+4*NL)+C2
      RETURN
   10 DP1=3.E7
      IF(BLEG(LEG+2*NL).EQ.0.)BLEG(LEG+2*NL)=.001
      GO TO 8
      ENTRY DR1W31
C        DYNAMIC RESISTANCE FOR 1 WAY RESTRICTOR   DATE 6/26/78
      M=IL
      LEG=BRANCHP(M+20*NBP2)
      Q=ABS(BLEG(LEG+2*NL))
      FI=BRANCHP(M+12*NBP2)
      IF(FI.LT.0.)GO TO 12
      C2=BRANCHP(M+5*NBP2)
      C3=BRANCHP(M+6*NBP2)
      IF(C2.GT..9)GO TO 11
```

6.2.3.7 (Continued)

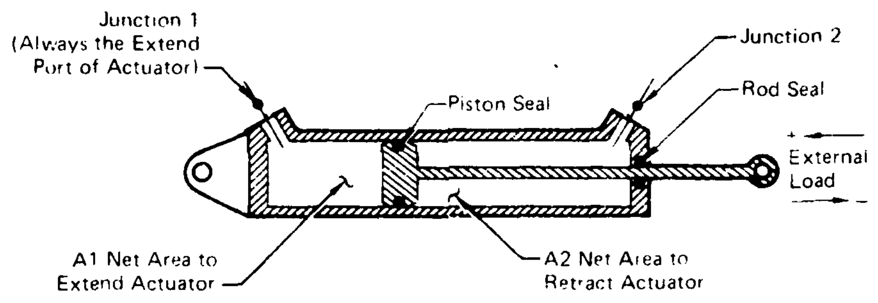
```
      IF(C2.EQ.0.)C2=.00001
      C4=C3/((1.-(C2/BRANCHP(M+3*NBP2))**4)**.5)
      DP1=(DENS*Q)/((236.*(C2**2)*C4)**2)
      GO TO 8
   11 CK=C2/(C3**2)
      DP1=CK*Q
      GO TO 8
   12 IF(BRANCHP(M+8*NBP2).EQ.0.)GO TO 13
      CK=BRANCHP(M+8*NBP2)/(BRANCHP(M+9*NBP2)**2)
      DP1=(BRANCHP(M+7*NBP2)/ABS(Q))+(CK*Q**2)
      GO TO 8
   13 C2=BRANCHP(M+7*NBP2)
      C3=BRANCHP(M+4*NBP2)
      GO TO 1
      ENTRY DVRE33
C        DYNAMIC RESISTANCE FOR RLF VALVE    DATE 6/13/78
      M=IL
      FI=BRANCHP(M+7*NBP2)
      C1=BRANCHP(M+3*NBP2)
      C2=BRANCHP(M+5*NBP2)
      IF(FI.LT.0.)GO TO 1
      DP1=3.E7
      GO TO 8
      END
```

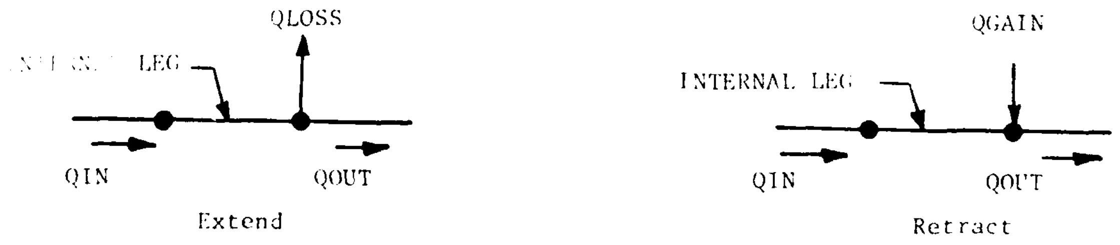
6.2.4 Subroutine DACT4

DACT4 is the dynamic element model for a simple actuator.

6.2.4.1 Math Model

The simple actuator is modeled as shown below, Figure 43. The flow is into either the extend or retract port. Piston leakage is calculated and also the net flow to operate the piston. The flow gain or loss depending on the piston direction is then calculated. The pressure rise or drop across the piston is also calculated based on the inlet pressure, the external load and the piston seal friction.





Branch Point Representation

FIGURE 43

SIMPLE ACTUATOR

6.2.4.2 Assumptions - Not applicable.

6.2.4.3 Computations

The actuator physical parameters are initialized from BRANCHP array. The piston leakage constant is calculated as

FCON = 200000.*(VISC/VI00)*(L/(3.14(6*PDIA))

For the actuator extending, the flow on the head side of the piston is

QP = QIN - QLEAK

Where the leakage flow is calculated from the previous pressure difference across the piston.

Q_{LEAK} = Pressure Drop/FCON.

The pressure on the opposite side of the piston is calculated as

PR = ((PH x Extend Area) - Load - Friction)/Retract Area

The flow out the return port is

QR = QP x (Retract Area/Extend Area)

The flow difference between flow in and flow out is input to PQL array.

PQL(N,2) = QR - QP

and the pressure difference across the piston is input into BLEG array column 5.

DP1 = PR - PH

BLEG(LEG,5) = BLEG(LEG,5) + DP1

Calculation for the actuator moving in the reverse direction is calculated in a similar manner as above with EXTA and RETA interchanged as well as QP, QR, PH and PR.

6.3.4.4 Approximations - The piston leakage resistance is approximated by 200000*(1./(3.1416*PDIA))

6.2.4.5 Limitations - Not applicable.

6.2.4.6 DACT4 Variable Names

Variables	Description	Dimension
ALOAD	Actuator load +compression - tension	LB
ALT	Altitude	FT
BLEG	Array containing calculation data	--
BRANCHP	Dynamic element data storage array	--
DENS	Fluid density at system temperature and atmospheric pressure	LB/FT^3
DPLOAD	Load equivalent pressure drop	PSID
DPT	Net pressure drop	PSID
DP1	Calculated ΔP across piston	PSI
D100	Density at 100°F	LB/FT^3
EXTA	Extend area	IN^2
FCON	Piston seal leakage constant	PSI/GPM
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
FRIC	Dynamic seal friction	LB
I	Integer counter	--
IL	Row location in BRANCHP array	--
JDIR	Piston motion direction indicator 1 = extend 2 = retract	--
LINT	Internal leg number	--
M	Row number in BRANCHP array	--
N	Branch point number of actuator	--
NBP2	Total number of rows in BRANCHP array	--

Variables	Description	Dimension
NL	Total number of rows in BLEG and ILEP arrays	--
NPQ	Total number of rows in PQL array	--
NR	Branch point number of retract side of actuator	--
PAMB	Atmospheric ambient pressure	PSI
PDIA	Piston diameter	IN
PE	Pressure on extend side of piston	PSI
PPOS	Piston position 0. = full retract	IN
PQL	Array containing pressure point data	--
PR	Pressure on retract side of piston	PSI
QIN	Flow into actuator	GPM
RETA	Piston retract area	IN^2
STR	Max stroke from full retract to full extend	IN
TEMP	Fluid temperature	°F
TY	Element type	--
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
VPORT	Pressure port junction number of valve controlling the actuator	--
V100	Fluid viscosity at 100°F	CENTISTOKES

6.2.4.7 <u>DACT4 Subroutine Listing</u>

```
      SUBROUTINE DACT4(IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
C       CALCULATES ACTUATOR PRESSURE DROP AND        DATE 12/1/79
C         FLOW GAIN OR LOSS FOR BRANCH POINT
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION BRANCHP(1),BLEG(1),PQL(1)
C       ROW NUMBER IN BRANCHP ARRAY
      M=IL
C       BRANCH POINT NUMBER OF ACTUATOR
      N=BRANCHP(M+3*NBP2)
      NR=BRANCHP(M+4*NBP2)
      EXTA=BRANCHP(M+5*NBP2)
      RETA=BRANCHP(M+6*NBP2)
      FRIC=BRANCHP(M+7*NBP2)
      ALOAD=BRANCHP(M+8*NBP2)
      STR=BRANCHP(M+9*NBP2)
      PPOS=BRANCHP(M+10*NBP2)
      PDIA=BRANCHP(M+11*NBP2)
      VPORT=BRANCHP(M+12*NBP2)
      LINT=BRANCHP(M+15*NBP2)
      IF(BRANCHP(M+3*NBP2).NE.0.)GO TO 1
      FCON=200000.*(VISC/V100)*(1./(3 1416*PDIA))
      BLEG(LINT+11*NL)=FCON
      RETURN
    1 DO 2 I=1,NBP2
      TY=BRANCHP(I)
      IF(VPORT.EQ.BRANCHP(I+2*NBP2).AND.TY.EQ.36.)GO TO 4
      IF(VPORT.EQ.BRANCHP(I+3*NBP2).AND.TY.EQ.35.)GO TO 4
      IF(VPORT.EQ.BRANCHP(I+3*NBP2).AND.TY.EQ.34.)GO TO 5
      IF(VPORT.EQ.BRANCHP(I+4*NBP2).AND.TY.EQ.34.)GO TO 6
    2 CONTINUE
      WRITE(6,3)VPORT
    3 FORMAT('DACT4 CAN NOT FIND PORT',F8.3,'IN BRANCHP ARRAY')
    4 JDIR=1
      IF(RETA.GT.EXTA)JDIR=2
      GO TO 7
    5 JDIR=1
      IF(BRANCHP(I+14*NBP2).EQ.2.)JDIR=2
      GO TO 7
    6 JDIR=2
      IF(BRANCHP(I+14*NBP2).EQ.2.)JDIR=1
C         FLOW INTO ACTUATOR
    7 DIN=ABS(BLEG(LINT+2*NL))
      PH=PQL(N)
      PR=PQL(NR)
      IF(PH.EQ.-1.)PH=2000.
      IF(PR.EQ.-1.)PR=3000.
      DPLOAD=ALOAD/EXTA
      IF(ALOAD.LT.0.)DPLOAD=ALOAD/RETA
      IF(JDIR.EQ.1)GO TO 8
      IF(JDIR.EQ.2)GO TO 9
```

6.2.4.7 (Continued)

```
  8 IF(PR.LT.0.001)PR=0.001
    PH=PR*(RETA/EXTA)
    DP1=PR-PH
    DPT=DP1-DPLOAD
    BLEG(LINT+4*NL)=DPT
    PQL(NR+NPQ)=(-((EXTA-RETA)/EXTA)*QIN)
    RETURN
  9 IF(PR.LT.0.001)PR=0.001
    PH=PR*(RETA/EXTA)
    DPT=DP1-DPLOAD
    BLEG(LINT+4*NL)=DPT
    PQL(N+NPQ)=((EXTA-RETA)/RETA)*QIN
    RETURN
    END
```

6.2.5 Subroutine DPUM5

DPUM5 is the dynamic element model for a variable delivery pump.

6.2.5.1 Math Model - The variable delivery hydraulic pump generates fluid flow in response to syste flow demand. The discharge pressure is a function of discharge flow. The steady state pump model, models the pump characteristic flows and pressures in the pump. The characteristic curves modeled, see Figure 44 , are:

(1) The standard flow versus pressure out curve

(2) The characteristic leakage from high pressure back to the pump case

(3) The leakage from the pump case back to the inlet

(4) The pump pressure out versus inlet pressure.

The pump case pressure is the pressure reference used for the pump outlet pressure. The case pressure in turn is referenced from atmospheric pressure through the reservoir and back to the pump case.

6.2.5.2 Assumptions - Not applicable.

6.2.5.3 Computations - The following equations were written with flows in gpm which is consistent with MIL-P-19692C.

The pump physical parameters are initialized from BRANCHP array. The flow out of the pump is initialized from BLEG array column 3. Using this flow, QT, the characteristic pressure out is calculated.

Flow < rated flow
POUT=P1-(P1-P2)x(QT/Q2)

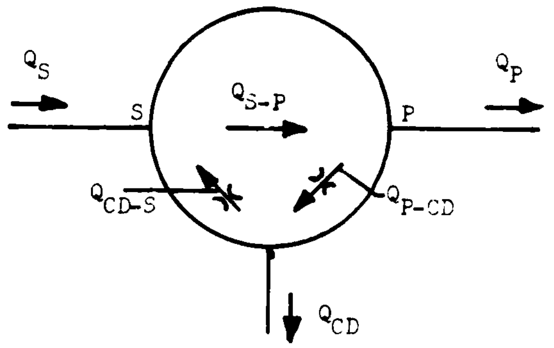
Flow > rated flow
POUT=P2-(P2-P3)x((QT-Q2)/(Q3-QZ))

The leakage flow from high pressure to the pump case is calculated as

QPCD=RCDLA-(.3x(QT/RQ))

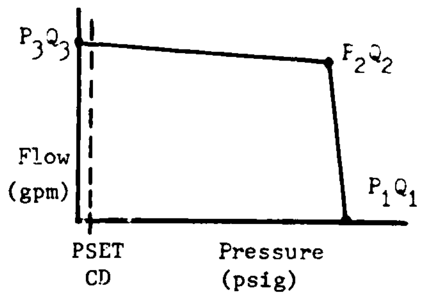
A default value of 1 gpm is used if no RCDLA value is input.

The pump case drain flow (flow out the port) is the leakage from high

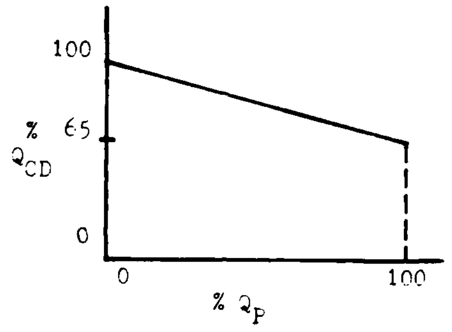


$Q_S = Q_P + Q_{CD}$

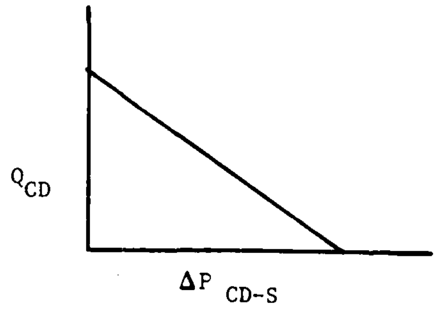
Pump flows



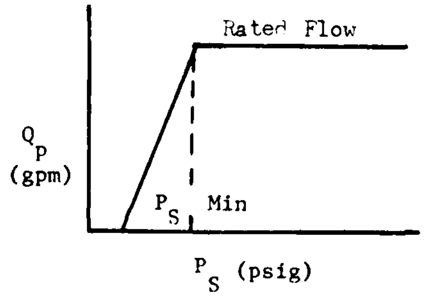
Flow out versus Pressure out



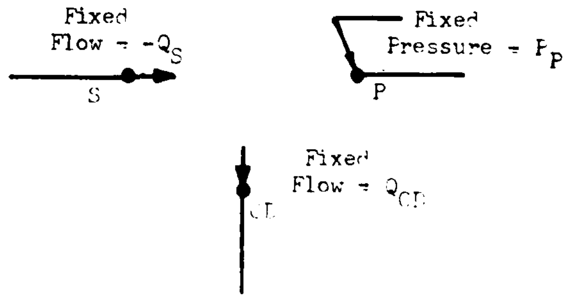
Case drain flow versus Flow out



Case drain flow versus ΔP between case and suction



Flow out versus Suction pressure



Branch point representation

FIGURE 44

VARIABLE DELIVERY PUMP

pressure to the case less the flow that leaks back to the inlet port. Therefore it is a function of QPCD and the pressure difference between the case and the inlet. The case drain flow

QCD=QPCDx(1.-(PC-PS)/RCDSDP)x(V100/VISC)

The default value for RCDSDP is 300 psid. This indicates that if the pressure difference between the pump case and inlet is greater than or equal to 300 psid, all the high pressure to case leakage flow goes back to the inlet.

The actual pump out pressure is calculated using the previously calculated POUT pressure from the characteristic curve and adjusting this to account for the actual pressure in the case less the case pressure at which POUT was set.

PP=POUT+PC-PSETA

The input values to the PQL array are 1 pressure and 2 flows.

Pump out pressure	PQL(MP3,1)=PP
Suction flow	PQL(MP2,2)=-QS
Case drain flow	PQL(MP1,2)=QCD

QS is calculated as QS=QT+QCD

The pump outlet pressure versus inlet pressure model is essentially a cavitating inlet which reduces the output flow.

6.2.5.4 Approximations - Not applicable.

6.2.5.5 Limitations - The model has an instability when iterating between values on each curve of the flow out versus pressure out graph in Figure 44. Care must be taken in choosing an operating point that is on one curve or the other.

6.2.5.6 DPUM5 Variable Names

Variables	Description	Dimension
ALT	Altitude	FT
BLEG	Array containing calculation data	--
BRANCHP	Dynamic element data storage array	--
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
D100	Weight density at 100°F	LB/FT^3
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
I	Integer counter	--
IL	Row location in BRANCHP array	--
ILEP	Array of leg numbers with the corresponding pressure on each end	--
LEG	Leg number (row in BLEG array)	--
MP1	Suction pressure point number	--
MP2	Pressure port pressure point number	--
MP3	Case drain port pressure point number	--
MP4	Upstream pressure point number for suction inlet leg	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG and ILEP arrays	--
NPQ	Total number of rows in PQL array	--
PAMB	Atmospheric ambient pressure	PSI
PC	Previous calculated pump case pressure	PSI
POLD	Previous calculated outlet pressure	PSI
POUT	Calculated pressure out of pump	PSI

Variables	Description	Dimension
POUTT	Calculated pressure out of pump	PSI
PP	Corrected pressure out of pump	PSI
PP2	Adjusted P2 pressure with reference to case pressure	PSI
PQL	Array containing pressure point data	--
PS	Previous calculated suction pressure	PSI
PSDEL	Suction pressure difference	PSID
PSET	Case pressure at which rated conditions are set	PSI
PSETA	Default for case set pressure	PSI
PSM	Minimum suction pressure	PSIG
PSMIN	Minimum suction pressure	PSIG
PSMINA	Default for minimum suction pressure	PSI
PSMINO	Minimum suction pressure	PSIA
PT	Absolute suction pressure	PSIA
P1	Pressure at 0 pump flow	PSI
P2	Pressure at pump rated flow	PSI
P3	Pressure with 0 system resistance	PSI
QCD	Flow out case drain port	GPM
QOLD	Previous calculated flow out of pump	GPM
QOUT	Pump pressure port flow rate	GPM
OPCD	Leakage flow from high pressure to case	GPM
QS	Inlet flow to pump	GPM
Q1	Flow with infinite system resistance	GPM
Q2	Rated pump flow	GPM
Q3	Flow with 0 system resistance	GPM

Variables	Description	Dimensions
RCDL	Default case drain flow	GPM
RCDLA	Rated case drain flow	GPM
RCDP	Rated pressure difference between case and suction	PSID
RCDSDP	Default pressure difference between case and suction	PSID
RPM	Pump RFM	RPM
RQ	Rated flow at rated rpm	GPM
RRPM	Rated pump rpm	RPM
TEMP	Fluid temperature	°F
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
V100	Fluid viscosity at 100°F	CENTISTOKES

6.2.5.7 DPUM5 Subroutine Listing

```
      SUBROUTINE DPUM5(IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ,ILEP)
C         CALCULATES PUMPOUT PRESSURE, CASE DRAIN FLOW
C         AND SUCTION FLOW    REV 12/01/79
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION ILEP(1),BRANCHP(1),BLEG(1),PQL(1)
      RPM=BRANCHP(IL+7*NBP2)
      IF(RPM.LE.0.)RETURN
      MP1=BRANCHP(IL+4*NBP2)
      MP2=BRANCHP(IL+5*NBP2)
      MP3=BRANCHP(IL+6*NBP2)
      LEG=BRANCHP(IL+16*NBP2)
      QOLD=ABS(BLEG(LEG+2*NL))
      BRANCHP(IL+17*NBP2)=QOLD
      POLD=PQL(MP2)
      RRPM=BRANCHP(IL+8*NBP2)
      RQ=BRANCHP(IL+9*NBP2)
      PSET=BRANCHP(IL+15*NBP2)
      PSETA=50.
      IF(PSET.GT.0.)PSETA=PSET
      P1=BRANCHP(IL+10*NBP2)
      P2=BRANCHP(IL+11*NBP2)
      PSMIN=BRANCHP(IL+12*NBP2)
      RCDP=BRANCHP(IL+13*NBP2)
      RCDL=BRANCHP(IL+14*NBP2)
      PC=PQL(MP3)
      IF(PC.EQ.-1.)PC=100.
      P3=0.
      Q1=0.
      Q2=RQ*RPM/RRPM
      BRANCHP(IL+18*NBP2)=Q2
      Q3=1.05*Q2
      PS=PQL(MP1)
      IF(PS.EQ.-1.)PS=50.
      PT=PS+PAMB
      RCDLA=1.
      IF(RCDL.GT.0.)RCDLA=RCDL
      QPCD=RCDLA-(.3*(QOLD/Q2))
      IF(QPCD.LT.0.)QPCD=0.
      RCDSDP=300.
      IF(RCDP.GT.0.)RCDSDP=RCDP
      QCD=QPCD*(1.-(PC-PS)/RCDSDP)*(V100/VISC)
      IF(QCD.LT.0.)QCD=0.
      PQL(MP3+NPQ)=QCD
      GO TO 1
      IF(BRANCHP(IL+19*NBP2).EQ.2.)GO TO 3
      IF(BRANCHP(IL+4*NBP2).EQ.0.)GO TO 1
      IF(BRANCHP(IL+4*NBP2).EQ.1.)GO TO 2
    1 QS=QOLD+QCD
      PQL(MP1+NPQ)=-QS
      POUT=P1-((P1-P2)*(QOLD/Q2))
```

6.2.5.7 (Continued)

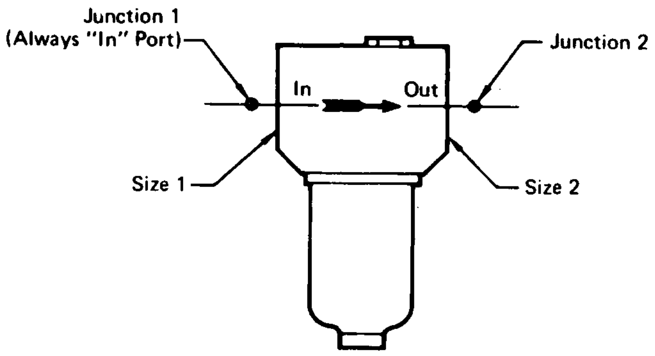
```
      POUTT=POUT+PC-PSETA
      IF(POUTT.LT.PC)POUTT=PC
      PQL(MP2)=POUTT
      GO TO 7
    2 PP=POLD-PC
      PP2=P2-PC
      QOUT=Q3-(PP/PP2)*(Q3-Q2)
      QS=QOUT+QCD
      PQL(MP1+NPQ)=-QS
      PQL(MP2+NPQ)=QOUT
      GO TO 7
    3 DO 4 I=1,NL
      IF(MP1.EQ.ILEP(I+NL))GO TO 5
    4 CONTINUE
    5 QS=BLEG(I+2*NL)
      IF(QS.LT.QCD)QCD=QS
      IF(QS.LT.0.)QCD=0.
      QOUT=QS-QCD+.000001
      IF(QS.LT.0.)QOUT=.000001
      PQL(MP2+NPQ)=QOUT
      PQL(MP3+NPQ)=QCD
      PSMINA=25.
      IF(PSMIN.GT.0.)PSMINA=PSMIN
      PSMIN0=10.
      PSDEL=PSMINA-PSMIN0
      IF(PSDEL.LT.1.)PSDEL=1.
      POUT=(QS/(Q3-Q1))*PSDEL+(PSMIN0-PAMB)
      IF(QS.GT.(Q3-Q1))POUT=PSMINA-PAMB
      IF(QS.LE.0.)POUT=PSMIN0-PAMB
      PSM=PSMIN0-PAMB
      IF(POUT.GT.PSM)GO TO 6
      MP4=(ILEP(I+NL))
      IF(POUT.GT.PQL(MP4))POUT=PQL(MP4)
    6 PQL(MP1)=POUT
    7 RETURN
      END
```

6.2.6 Subroutine DFIL6

DFIL6 is the dynamic element model for a hydraulic filter assembly.

6.2.6.1 Math Model

The filter model may be either a bypass type or non-bypass type, see Figure 45. The effect of the filter port frictional resistance is calculated in SCONST2. SCONST2 also calculates the energy loss due to the sudden expansion into the volume and the sudden expansion out of the volume. The loss due to the filter element is calculated by DFIL6.



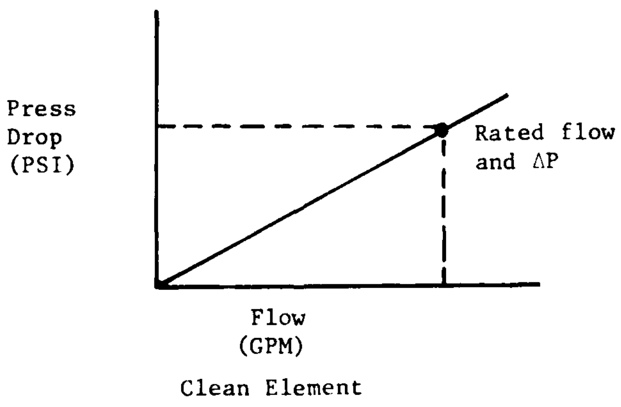


FIGURE 45
FILTER

6.2.6.2 Assumptions - Not applicable.

6.2.6.3 Computations

The input data parameters are initialized from BRANCHP array. A test is made to determine if flow is through the filter in the normal flow direction. If the flow is reversed the pressure drop is higher than the normal flow direction. This pressure drop is approximated by

PDT = (50./D1) x Q

In the normal flow direction, the fluid volume is converted to an equivalent area for calculation of the expansion and contraction energy loss. These values are converted to pressure drop in psi. The element pressure drop is calculated from the input clean element ratings. The pressure drop is corrected for viscosity where

PDEL = CDP x (Q/CDQ) x (VISCP/RVIS)

If a contaminated element factor is used,

CF = (1.-CONTAM)

Then the pressure drop term becomes

PDT = PDEL/CF

A test is next made to see if the PDT pressure drop exceeds the bypass allowed pressure drop. Using an equivalent orifice flow equation, the element and the bypass are calculated as parallel flow orifices.

6.2.6.4 Approximations - Not applicable.

6.2.6.5 Limitations - Not applicable.

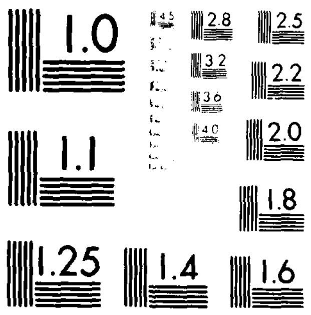
6.2.6.6 DFIL6 Variable Names

Variable	Description	Dimension
A	Flow direction multiplication factor	--
ALT	Altitude	FT
BLEG	General purpose array	--
BPD	Bypass pressure drop	--
BRANCHP	Dynamic element data storage array	--
CDP	Filter element rated pressure drop	PSI
CDQ	Rated flow of clean element	GPM
CF	Contamination factor	--
CONTAM	Contamination factor	--
DENP	Fluid weight density at system pressure	LB/FT^3
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DI	Assembly direction indicator	--
DP1	Pressure drop through filter element	PSI
DRPD	Difference between bypass pressure drop and rated pressure drop	PSID
D1	Port 1 diameter	IN
D100	Weight density at 100°F	LB/FT^3
D2	Port 2 diameter	IN
EQD	Equivalent diameter	IN
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
FV	Filter assembly fluid volume	IN^3
F2	Intermediate calculation of bypass flow rate	--

Variable	Description	Dimension
IL	Row location in BRANCHP array	--
LEG	Leg number	--
M	Row in BRANCHP array	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG array	--
PAMB	Atmospheric ambient pressure	--
PDEL	Actual filter element pressure drop	PSI
PDT	Total pressure drop	PSI
PDL	Entrance pressure drop	PSI
PD2	Exit pressure drop	PSI
PEL	Corrected filter element pressure drop with contamination	--
Q	Leg flow	GPM
Q1	Calculated flow through filter element	GPM
Q2	Calculated flow through bypass relief	GPM
RK1	Calculated resistance factor for rated pressure drop	--
RK2	Calculated resistance factor for DRPD	--
RPD	Rated pressure drop	PSI
RVIS	Rated viscosity	CENTISTOKES
SKBS	Function to calculate exit pressure drop	--
SKSB	Function to calculate entrance pressure drop	--
TEMP	Fluid temperature	°F
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
V100	Fluid viscosity at 100°F	CENTISTOKES

AD-A089 240 MCDONNELL AIRCRAFT CO ST LOUIS MO F/G 1/3
AIRCRAFT HYDRAULIC SYSTEMS DYNAMIC ANALYSIS. VOLUME VI. STEADY --ETC(U)
APR 80 R LEVEK, B YOUNG F33615-74-C-2016
UNCLASSIFIED AFAPL-TR-76-43-VOL-6 NL

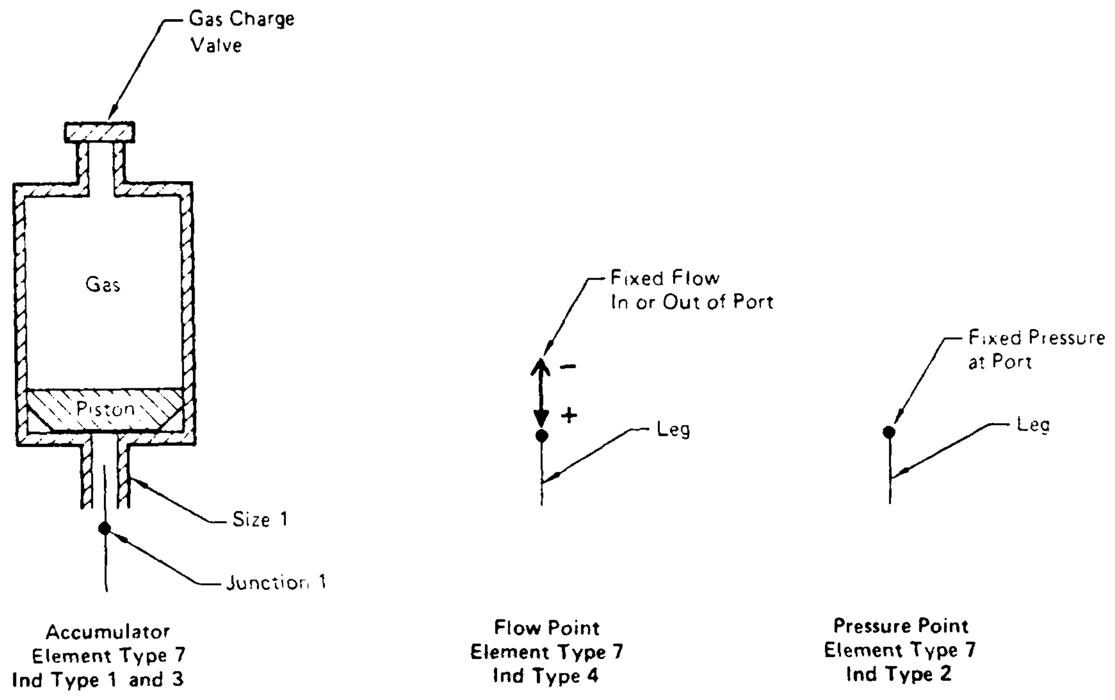
3 OF 4
AD
A089240


1.0
4.5
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.1
1.8
1.25
1.4
1.6

6.2.6.7 DFIL6 Subroutine Listing

```
      SUBROUTINE DFIL6(IL,BRANCHP,NBP2,BLEG,NL)
C        DYNAMIC RESISTANCE FOR FILTER     DATE 12/1/79
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION BRANCHP(1),BLEG(1)
      M=IL
      LEG=BRANCHP(IL+20*NBP2)
      D1=BRANCHP(M+3*NBP2)
      D2=BRANCHP(M+4*NBP2)
      FV=BRANCHP(M+5*NBP2)
      CDQ=BRANCHP(M+6*NBP2)
      CDP=BRANCHP(M+7*NBP2)
      RVIS=BRANCHP(M+8*NBP2)
      CONTAM=BRANCHP(M+9*NBP2)
      RPD=BRANCHP(M+10*NBP2)
      BPD=BRANCHP(M+11*NBP2)
      DI=BRANCHP(M+12*NBP2)
      Q=BLEG(LEG+2*NL)
      A=1.
      IF(Q.GE.0..AND.DI.EQ.1.)GO TO 2
      IF(Q.LT.0..AND.DI.EQ.2.)GO TO 1
      IF(Q.LT.0..AND.DI.EQ.1.)A=-1.
      GO TO 3
    1 A=-1.
    2 EQD=1.2407*FV**(1./3.)
      PD1=SKSB(D1,EQD,DENP)*Q**2
      PD2=SKBS(EQD,D2,DENP)*Q**2
      PDEL=CDP*(Q/CDQ)*(VISC/RVIS)
      CF=.0001
      IF(CONTAM.LT.1.)CF=(1.-CONTAM)
      PEL=PDEL/CF
      PDT=PEL+PD1+PD2
      IF(BPD.EQ.0.)GO TO 4
      IF(PEL.LE.RPD)GO TO 4
      DRPD=BPD-RPD
      RK2=(DRPD/(CDQ*CDQ))*((VISC/RVIS)**.25)*(DENP/DENS)
      RK1=(CDP/CDQ)*(VISC/RVIS)*(DENP/DENS)/CF
      F2=SQRT(((RK1/RK2)**2)-4.*((RPD-RK1*Q)/RK2))
      Q2=(-(RK1/RK2)+F2)/2.
      Q1=Q-Q2
      DP1=RK1*Q1
      PDT=(DP1+PD1+PD2)*A
      GO TO 4
    3 PDT=(50./D1)*Q*A
    4 BLEG(LEG+11*NL)=BLEG(LEG+11*NL)+ABS(PDT/Q)
      RETURN
      END
```

6.2.7 Subroutine DACC7


Gas Charge Valve
Gas
Piston
Size 1
Junction 1
Accumulator Element Type 7 Ind Type 1 and 3
Fixed Flow In or Out of Port
−
+
Leg
Flow Point Element Type 7 Ind Type 4
Fixed Pressure at Port
Leg
Pressure Point Element Type 7 Ind Type 2

ELEMENT TYPE 7 - ACCUMULATOR

GP79-0981-33

FIGURE 46

The type 7 accumulator is used also for fixed flows and fixed pressures; see Figure 46. Subtype 1. is a passive (static) accumulator where no flow is in the leg to the accumulator and the accumulator pressure is the same as the pressure at the point where the leg branches off to go to the accumulator. Subtype 2. is a fixed pressure point and maintains this pressure regardless of the flow in the leg leading to or away from the point. This point has to be an end point in the system and cannot be an intermediate point between 2 elements. Subtype 3. is a dynamic accumulator. An accumulator may be a piston type, bladder type, or an air to oil type. Any of these types can be input as Type 7 because an accumulator power capability is based on the precharge gas volume, fluid minimum and maximum volume and initial pressure or initial fluid volume are used only when the dynamic accumulator is used. Subtype 4. is a fixed flow point with similar restrictions as the Subtype 2. This point will maintain the constant flow regardless of system conditions or changes in system conditions. When a call to DACC7 subroutine is made the program flow is routed to the appropriate section through the subtype indicator.

SUMMARY OF ELEMENT TYPE 7. SUBTYPE OPTIONS

SUBTYPE		DESCRIPTION
1.	STATIC PRESSURE POINT	PRESSURE WILL BE CALCULATED AT PRESSURE POINT
2.	FIXED PRESSURE POINT	SETS PRESSURE AT PRESSURE POINT
3.	DYNAMIC PRESSURE POINT	PRESSURE AND/OR VOLUME IS SET AT PRESSURE POINT AND UPDATED DURING PROGRAM CALCULATION
4.	FIXED FLOW POINT	FLOW IS SET AT PRESSURE POINT

6.2.7.1 DACC7 Variable Names

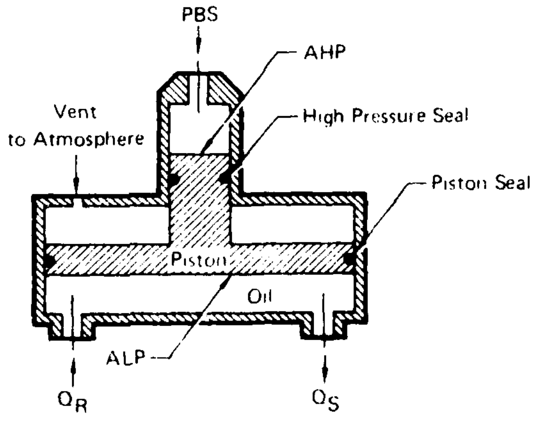
Variables	Description	Dimensions
BRANCHP	Array containing dynamic element data	--
IL	Row location in BRANCHP array	--
J	Element sub-type 1 Static accumulator 2 Fixed pressure point 3 Dynamic accumulator 4 Fixed flow point	--
N	Pressure point number	--
NBP2	Total number of rows in BRANCHP array	--
NPQ	Total number of rows in PQL array	--
PQL	Array of pressure points, flows and port numbers	--

6.2.7.2 DACC7 Subroutine Listing

```
      SUBROUTINE DACC7(IL,BRANCHP,NBP2,PQL,NPQ)
C       DYNAMIC ACCUMULATOR--FIXED FLOW OR PRESSURE POINT
C                     12/01/79
      DIMENSION BRANCHP(1),PQL(1)
      N=BRANCHP(IL+2*NBP2)
      J=BRANCHP(IL+3*NBP2)
      GO TO (4,1,2,3),J
    1 PQL(N)=BRANCHP(IL+4*NBP2)
      RETURN
    2 PQL(N)=BRANCHP(IL+10*NBP2)
      RETURN
    3 PQL(N+NPQ)=BRANCHP(IL+4*NBP2)
    4 RETURN
      END
```

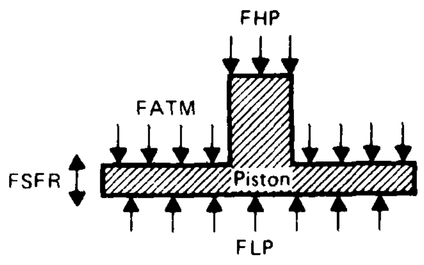
6.2.8 Subroutine DRESV

DRESV is the dynamic subroutine for calculating the reservoir internal pressure and bootstrap flow. Type 9 flow through reservoir type 91 appendix reservoir and type 92 constant pressure reservoir are defined in this subroutine. A type 9 reservoir, Figure 47, flows include the flow in the return port and the flow out of the suction port to the pump. The flow in the bootstrap line is dependent on the difference between the flow in and flow out. Since the type 91 reservoir has only one low pressure port, the flow in the bootstrap line is directly dependent on this one flow. The calculated internal pressure is dependent on the bootstrap pressure and is also referenced to the external atmospheric pressure. For a type 92 reservoir, the internal pressure is held constant by an independent source.



PATM Atmospheric Pressure (psi)
AHP - High Pressure Area (in.**2)
ALP Low Pressure Area (in.**2)
FR Frictional Force (lb)
QR - Return Flow (gpm)
QS - Suction Flow to Pump (gpm)
PBS Bootstrap Pressure (psi)
PR Reservoir Pressure (psi)
ALT Altitude (ft)
PALT - Pressure at Altitude (psi)
PDEL - Pressure Difference at Altitude (psi)
FLP - Force of Internal Reservoir Pressure (lb)
FHP Force of Bootstrap Pressure (lb)
FATM Force of Atmospheric Pressure (lb)
FSFR - Equivalent Force Resistance of Seal Friction (lb)

Piston Force Balance



Pressure at Altitude

$$PALT = \left[PATM^{0.192} \left(1 - \frac{0.0036 \times ALT}{519} \right) \right]^{5.208}$$

Force Balance

$FLP = FHP + FATM \pm FSFR$

$FLP = PR \times ALP$

$FHP = PBS \times AHP$

$FATM = (ALP - AHP) \times PALT$

$FSFR = 10 \times (\text{Sum of Dynamic Seal Ods})$

$PDEL = PATM - PALT$

$$PR = \frac{(PBS \times AHP - (PDEL \times (ALP - AHP) \pm FR)}{ALP}$$

FIGURE 47
FLOW THROUGH RESERVOIR

6.2.8.1 Math Model - The reservoir model includes the parameters shown in Figure 47. The model is basically a force balance on the reservoir piston. The direction of piston motion is obtained by comparing flow in and flow out for the flow through reservoir and the flow direction for the appendix reservoir. When the piston direction is determined, the flow direction for the bootstrap is determined.

6.2.8.2 Approximations - Not applicable.

6.2.8.3 Computations - The reservoir physical parameters are initialized from BRANCHP array. For an appendix reservoir, an energy loss term for the flow in or out of the volume is calculated because the flow direction in the appendix line is not fixed.

Referencing to a standard atmosphere, the pressure at altitude is calculated and PDEL is

$$\mathrm{PDEL}=\mathrm{PATM}-\mathrm{PALT}$$

The previous iteration bootstrap pressure is used in calculating the internal reservoir pressure. Based on the piston direction, the friction term always opposes the direction of motion.

The internal reservoir pressure is calculated as

$$\mathrm{PR}=(\mathrm{PBS}\times\mathrm{AHP}-(\mathrm{PDEL}\times(\mathrm{ALP}-\mathrm{AHP}))\pm\mathrm{FR})/\mathrm{ALP}$$

The calculated reservoir internal pressure is input to PQL array to be used as a fixed pressure for the iteration. The same pressure is used for the flow through reservoir suction pressure at the reservoir. A calculation is made for the flow in the bootstrap line as

$$\mathrm{QL}=(\mathrm{QR}-\mathrm{QS})\times(\mathrm{AHP}/\mathrm{ALP})$$

For an appendix reservoir QS=0. This value is input to PQL array column 2 as a flow gain or loss.

6.2.8.4 Assumptions - Not applicable.

6.2.8.5 Limitations - Not applicable.

6.2.8.6 DRESV Variable Names

Variable	Description	Dimension
AHP	High pressure area	IN^2
ALP	Low pressure area	IN^2
ALT	Altitude	FT
BLEG	General purpose array	--
BRANCHP	Dynamic element data storage array	--
DENP	Fluid weight density at system pressure	LB/FT^3
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DIAO	Diameter of previous element	IN
DIAR	Reservoir piston diameter	IN
DP	Energy loss pressure drop	PSI
D100	Weight density at 100°F	LB/FT^3
EL	Energy loss coefficient	PSI/GPM^2
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
FR	Piston friction force	LB
IL	Row location in BRANCHP array	--
J1	Bootstrap pressure point number	--
J2	Return/suction pressure point number	--
LR	Return leg number	--
M1	Bootstrap pressure point number	--
M2	Return port leg number	--
M3	Suction port leg number	--
M4	Return port pressure point number	--

Variable	Description	Dimension
M5	Suction port pressure point number	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG array	--
NPQ	Total number of rows in PQL array	--
PALT	Atmospheric pressure at altitude	PSIA
PAMB	Atmospheric ambient pressure	PSI
PATM	Atmospheric pressure	PSI
PBS	Bootstrap pressure	PSI
PDEL	Difference between standard and atmospheric pressure	PSI
PQL	Array of pressure points, flows and port numbers	--
PR	Internal reservoir pressure	PSI
PRO	Previous calculated pressure	PSI
PRD	Previous calculated reservoir pressure	PSI
QL	Leakage flow rate in bootstrap pressure line	GPM
QR	Return port flow	GPM
QS	Suction port flow	GPM
SKBS	Function to calculate exit pressure drop	--
SKSB	Function to calculate entrance pressure drop	--
TEMP	Fluid temperature	°F
TY	Element type	--
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
V100	Fluid viscosity at 100°F	CENTISTOKES

6.2.8.7 <u>DRESV Subroutine Listing</u>

```
      SUBROUTINE DRESV(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
C         CALCULATES DYNAMIC RESISTANCE AND INTERNAL
C         RESERVOIR PRESSURE    DATE 12/01/79
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION BLEG(1),PQL(1),BRANCHP(1)
      IF(TY.EQ.91.)GO TO 1
      IF(TY.EQ.92.) GO TO 11
      FR=BRANCHP(IL+9*NBP2)
      M1=BRANCHP(IL+5*NBP2)
      PBS=PQL(M1)
      IF(PBS.EQ.(-1.))PBS=3000.
      ALP=BRANCHP(IL+8*NBP2)
      AHP=BRANCHP(IL+7*NBP2)
      M2=BRANCHP(IL+10*NBP2)
      M3=BRANCHP(IL+12*NBP2)
      QR=BLEG(M2+2*NL)
      QS=BLEG(M3+2*NL)*BRANCHP(IL+13*NBP2)
      GO TO 5
    1 QS=0.
      ALP=BRANCHP(IL+6*NBP2)
      AHP=BRANCHP(IL+5*NBP2)
      LR=BRANCHP(IL+9*NBP2)
      QR=BLEG(LR+2*NL)
      DIAO=BLEG(LR+3*NL)
      FR=BRANCHP(IL+7*NBP2)
      J1=BRANCHP(IL+4*NBP2)
      PBS=PQL(J1)
      J2=BRANCHP(IL+3*NBP2)
      PRO=PQL(J2)
      IF(PRO.EQ.-1.)PRO=50.
      IF(PBS.EQ.(-1.))PBS=3000.
      DIAR=(ALP/.7854)**.5
      IF(QR)2,5,3
    2 EL=SKBS(DIAR,DIAO,D100)
      GO TO 4
    3 EL=SKSB(DIAO,DIAR,D100)
    4 DENP=(1.+PRO/200000.)*DENS
      DP=EL*(DENP/D100)*QR*QR
      BLEG(LR+4*NL)=BLEG(LR+4*NL)-DP
    5 CONTINUE
      PATM=14.6999
      PALT=PAMB
      PDEL=PATM-PALT
      IF(ABS(PDEL).LT..001)PDEL=0.
      IF(QR-QS)6,7,8
    6 PR=(PBS*AHP-(PDEL*(ALP-AHP))-FR)/ALP
      GO TO 9
    7 PR=(PBS*AHP-(PDEL*(ALP-AHP)))/ALP
      GO TO 9
    8 PR=(PBS*AHP-(PDEL*(ALP-AHP))+FR)/ALP
```

6.2.8.7 (Continued)

```
  9 CONTINUE
    IF(TY.EQ.91)GO  TO 10
    QL=(QR-QS)*(AHP/ALP)
    PQL(M1+NPQ)=QL
    M4=BRANCHP(IL+4*NBP2)
    M5=BRANCHP(IL+6*NBP2)
    PQL(M4)=PR
    PQL(M5)=PR
    RETURN
 10 PQL(J2)=PR
    PQL(J1+NPQ)=QR*(AHP/ALP)
    RETURN
 11 CONTINUE
    M1=BRANCHP(IL+3*NBP2)
    PQL(M1)=BRANCHP(IL+5*NBP2)
    M1=BRANCHP(IL+4*NBP2)
    PQL(M1)=BRANCHP(IL+5*NBP2)
    RETURN
    END
```

6.2.9 DPEC10 Subroutine

The DPEC10 subroutine is used in SSFAN to allow the use of special elements in a hydraulic system that cannot be described by the other element subroutines. Values of flow versus pressure drop for an element at a specific temperature and viscosity are read into the BRANCHP array. The DPEC10 subroutine, in conjunction with INTERP, returns a pressure drop for a given element, temperature, viscosity and flow.

6.2.9.1 Math Model

The maximum number of data points for flow vs PD is six at a temperature and viscosity. Extrapolation for values greater than the data points are allowed. The pressure drop for the element is corrected for viscosity before it is returned to the main program by the following equation.

$$PD_{\text{Program}} = PD_{\text{From DPEC10}} \left[\frac{VISC_{\text{Program}}}{VISC_{\text{Element}}} \right]^{.25} \qquad (1)$$

6.2.9.2 Assumptions - Not Applicable.

6.2.9.3 Computations

The DPEC10 subroutine is called from TTL subroutine. The DARR subroutine is called to place the viscosity and temperature data for the special element into two arrays FARR1 and FARR2. For only two data points linear interpolation is used and IDEG is set to 10. Otherwise the call to INTERP for the elements current flow in the leg (Q) contains a second degree interpolation. This pressure drop from INTERP is viscosity corrected by equation (1) and program control is transferred to TTL.

6.2.9.4 Approximations - Not applicable.

6.2.9.5 Limitations

DPEC10 is a versatile subprogram that allows the addition of any specially made element in a hydraulic system, if the element is modeled correctly.

6.2.9.6 DPEC10 Variable Names

Variable	Description	Dimension
ALT	Altitude	FT
BLEG	General purpose array	--
BRANCHP	Dynamic element data storage array	--
CK	Calculated resistance coefficient for single data point input	--
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DP	Pressure drop corrected for viscosity effects	PSI
DP1	Pressure drop uncorrected for viscosity effects	PSI
D100	Weight density at 100°F	LB/FT^3
FARR1	Dummy array for flow data	--
FARR2	Dummy array for pressure drop data	--
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
IDEG	Degree of interpolation	--
IL	Row location in BRANCHP array	--
IND	Solution indicator = 0 Normal interpolation = 1 Extrapolation outside of data range	--
K	Integer row number in BRANCHP array	--
LEG	Leg number	--
M	Number of data points	--

Variable	Description	Dimension
M1	Locator for BRANCHP to determine if linear interpolation should be used	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG array	--
PAMB	Atmospheric ambient pressure	PSI
Q	Leg flow rate	GPM
RP	Rated pressure drop for single data point input	PSI
RQ	Rated flow for single data point input	GPM
TEMP	Fluid temperature	°F
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
V100	Fluid viscosity at 100°F	CENTISTOKES

6.2.9.7 DPEC10 Subroutine Listing

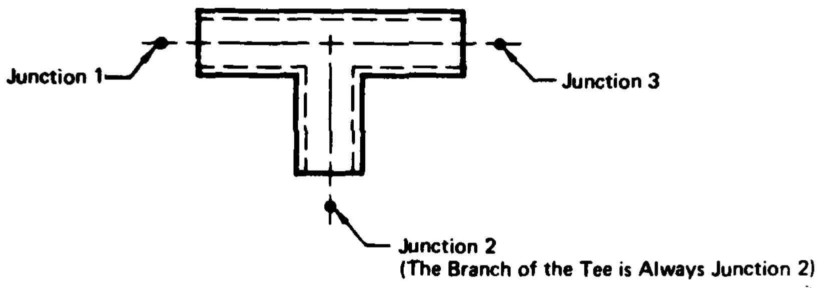
```
      SUBROUTINE DPEC10(IL,BRANCHP,NBP2,BLEG,NL)
C     DYNAMIC RESISTANCE FOR SPECIAL ELEMENT    DATE 6/28/78
      DIMENSION BRANCHP(1),BLEG(1)
      DIMENSION FARR1(6),FARR2(6)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DATA FARR1,FARR2/12*0./
      K=IL
      LEG=BRANCHP(K+20*NBP2)
      M=BRANCHP(K+6*NBP2)
      M1=7+(2*M)
      Q=ABS(BLEG(LEG+2*NL))+.000001
      IDEG=20
      IF(BRANCHP(K+M1*NBP2).GT.0..AND.M.LT.6)IDEG=10
      IF(M.EQ.1)GO TO 2
      IF(M.EQ.2)IDEG=10
      CALL DARR(FARR1,FARR2,M K,BRANCHP,NBP2)
      CALL INTERP(Q,FARR2,FARR1,IDEG,M,DP1,IND)
    1 DP=DP1*((VISC/BRANCHP(K+5*NBP2))**.25)
      IF(IDEG.EQ.10)DP=DP1*(VISC/BRANCHP(K+5*NBP2))
      DP=DP/Q
      BLEG(LEG+11*NBP2)=BLEG(LEG+11*NBP2)+DP
      RETURN
    2 RP=BRANCHP(K+7*NBP2)
      RQ=BRANCHP(K+8*NBP2)
      CK=RP/RQ**2
      DP1=CK*Q*Q
      IF(IDEG.EQ.10)DP1=(RP/RQ)*Q
      GO TO 1
      END
```

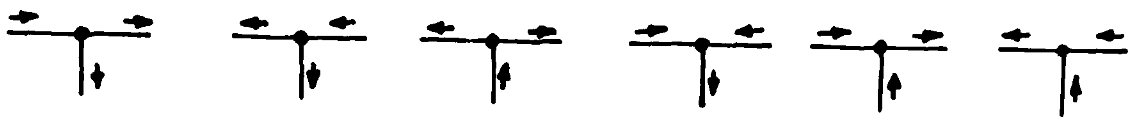
6.2.10 Subroutine DTEE24

DTEE24 is the dynamic element model for a tee.

6.2.10.1 Math Model

When DTEE24 is called from TTL subroutine, the connecting ports to the tee are initialized for the assembly direction and connecting leg numbers. The flow into each port and the port area and fluid velocity are also calculated. A test is made for flow direction, then the energy loss for combining or dividing flow is calculated. Figure 48 shows the tee and its branch point representation.





The Six Flow Conditions for a Tee in DTEE24

FIGURE 48

TEE

6.2.10.2 Assumptions - See Appendix C.

6.2.10.3 Computations

The method of derivation for the tee and cross element flow dividing and flow combining energy losses is shown in Appendix C.

6.2.10.4 Approximations - See Appendix C.

6.2.10.5 Limitations - Not applicable.

6.2.11 Entry DCR025

DCR025 is called from TTL and is similar to DTEE24. The connecting ports are initialized for assembly direction and connecting leg numbers. The previous calculated flow rates are initialized for the flow rate and flow direction in each leg. Using this information the energy loss for combining or dividing flow is calculated similar to the tee calculation as shown in Appendix C. There are fourteen flow conditions for the cross. The cross and tee are both represented as one branch point during dynamic calculation.

6.2.11.1 DTEE24 - DCR025 Variable Names

Variable	Description	Dimension
A	Port area	IN^2
ALT	Altitude	FT
B	Array for marking beginning or end of leg at junction	--
BLEG	Array containing calculation data	--
BRANCHP	Array containing dynamic element data	--
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DI	Flow direction indicator	--
DP	Calculated pressure drop	PSI
D100	Weight density at 100°F	LB/FT^3
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
I	Integer counter	--
IL	Row location in BRANCHP array	--
M	Integer counter	--
N	Array for leg numbers	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG array	--
PAMB	Atmospheric ambient pressure	PSI
Q	Fluid flow rate (see Appendix C)	GPM
QO	Fluid flow rate (see Appendix C)	GPM
Q1	Fluid flow rate (see Appendix C)	GPM

Variable	Description	Dimension
Q2	Fluid flow rate (see Appendix C)	GPM
TEMP	Fluid temperature	°F
V	Fluid velocity (see Appendix C)	FT/SEC
VO	Fluid velocity (see Appendix C)	FT/SEC
V1	Fluid velocity (see Appendix C)	FT/SEC
V2	Fluid velocity (see Appendix C)	FT/SEC
V100	Fluid viscosity at 100°F	CENTISTOKES

6.2.11.2 DTEE24 - DCR025 Subrouting Listing

```
      SUBROUTINE DTEE24(IL,BRANCHP,NBP2,BLEG,NL)
C           DYNAMIC RESISTANCE FOR TEE     REV 12/10/79
      DIMENSION N(3),DI(3),Q(3),A(3),V(3)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION BRANCHP(1),BLEG(1)
      DO 1 I=1,3
      N(I)=BRANCHP(IL+(6+I)*NBP2)
      B=BRANCHP(IL+(9+I)*NBP2)*2.-1.
      Q(I)=BLEG(N(I)+2*NL)
      A(I)=(BRANCHP(IL+(3+I)*NBP2)**2)*.7854
      V(I)=((ABS(Q(I))*231./60.)/A(I))/12.
      DI(I)=B*Q(I)
      Q(I)=ABS(Q(I))
    1 CONTINUE
      IF(DI(1)*DI(2)*DI(3))2,11,14
    2 IF(DI(2).GT.0.)GO TO 3
      GO TO 6
    3 IF(DI(1).GT.0.)GO TO 4
       GO TO 5
    4 V0=V(3)
      V1=V(1)
      V2=V(2)
      GO TO 7
    5 V0=V(1)
      V1=V(3)
      V2=V(2)
      GO TO 7
    6 V0=V(2)
      V1=V(1)
      V2=V(3)
    7 DP=.68*V0**2+.18*V1**2+.55*V2**2-.36*V0*V1-.2097*V0*V2
      GO TO 9
    8 DP=V0**2+.55*V1**2+.55*V2**2-.2097*V0*(V1+V2)
    9 DP=(DP*DENS)/(144.*32.2)
      DO 10 M=1,3
      IF(DI(M).LT.0.)BLEG(N(M)+4*NL)=BLEG(N(M)+4*NL)-DP
   10 CONTINUE
      RETURN
   11 IF(DI(2).EQ.0.)RETURN
      IF(DI(2).GT.0.)GO TO 12
      V0=V(2)
      V1=V(1)
      IF(DI(1).EQ.0.)V1=V(3)
      GO TO 13
   12 V0=V(1)
      IF(DI(1).EQ.0.)V0=V(3)
      V1=V(2)
   13 DP=.5*V0**2+.55*V1**2-.2097*V0*V1
      GO TO 9
   14 IF(DI(2).LT.0.)GO TO 15
```

6.2.11.2 (Continued)

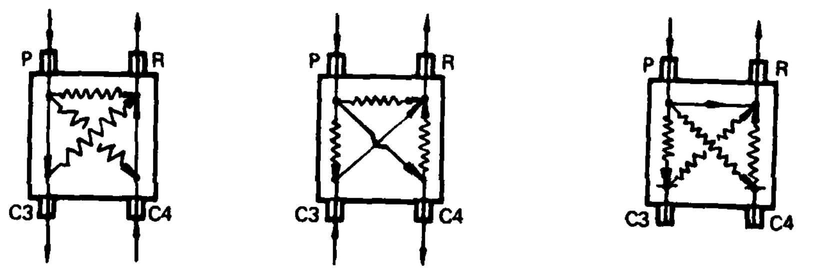
```
      GO TO 18
   15 IF(DI(1).LT.0.)GO TO 16
      GO TO 17
   16 V0=V(3)
      V1=V(2)
      V2=V(1)
      Q0=Q(3)
      Q1=Q(2)
      Q2=Q(1)
      GO TO 19
   17 V0=V(1)
      V1=V(2)
      V2=V(3)
      Q0=Q(1)
      Q1=Q(2)
      Q2=Q(3)
      GO TO 19
   18 V0=V(2)
      V1=V(1)
      V2=V(3)
      Q0=Q(2)
      Q1=Q(1)
      Q2=Q(3)
      GO TO 20
   19 DP=.3*V1**2+.48*V2**2+V0**2-2.*V0*((V2*Q2+V1*Q1)/Q0)
      GO TO 9
   20 DP=.3*V1**2+.3*V2**2+V0**2-.466*V0*((V1*Q1+V2*Q2)/Q0)
      GO TO 9
      ENTRY DCH025
      RETURN
      END
```

6.2.12 Subroutine DVSO34

DVSO34 is the dynamic subroutine for calculating the internal resistance or losses for a type 34 (four way three position valve), a type 35 (three-way-two position valve), a type 36 (2 way-two position valve), a type 37 (flow regulator), and a type 38 (orifice sizer).

6.2.12.1 Math Model

The type 34 valve model is shown in Figure 49. Five internal flow paths are considered in the valve. There are three positions identified as control code 1, 2 and 3.



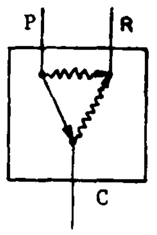
CONTROL CODE 1 CONTROL CODE 2 CONTROL CODE 3

TYPE 34 VALVE

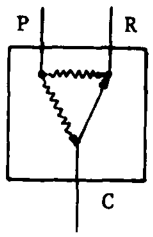
Figure 49

Using control code 1, pressure is ported to C3 and C4 is ported to return. Leakage flow is from pressure to return and high resistance are placed in the legs from P to C4 and C3 to R. Leakage flow is considered to be laminar and normal flow through the valve is considered turbulent. When control code 2 is used, pressure is ported to C4 and C3 is ported to return. High resistances are placed in the legs from P to C3 and C4 to return. With control code 3 the valve is closed with high resistances placed in all legs. Leakage flow is considered to be from P to R and may be input by the user.

The type 35 valve, Figure 50, has three internal flow paths and two control codes (1 and 3).



CONTROL CODE 1

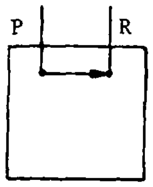


CONTROL CODE 3

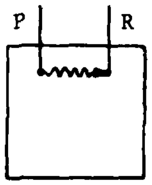
TYPE 35 VALVE
Figure 50

Control Code 1 ports pressure to C and Control Code 3 ports C to Return.

The type 36 valve, Figure 51, has only one internal flow path, but has two control codes (1 and 3). It is a simple on-off type valve.



CONTROL CODE 1



CONTROL CODE 3

TYPE 36 VALVE
Figure 51

The type 37 flow regulator and type 38 orifice sizer are similar, except the orifice sizer calculates an orifice diameter through FLOCHEK subroutine. The flow regulator model, orifice sizer model, Figure 52, places a high resistance in the internal leg and uses a Q_{loss} and Q_{gain} term for the fixed flow rate. If the pressure drop is less than the minimum pressure drop, the Q_{loss} and Q_{gain} terms are set to zero and a low resistance is placed in the internal leg.

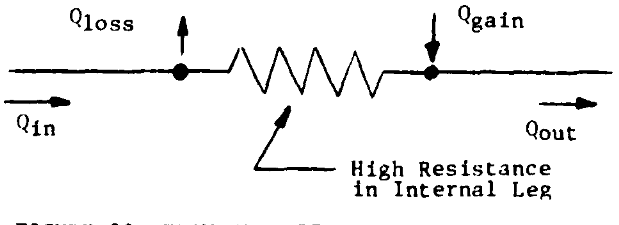


FIGURE 52 FLOW TYPE 37 AND 38 REGULATOR AND ORIFICE SIZER

6.2.12.2 Assumptions - Not applicable.

6.2.12.3 Computations -

The valve physical and flow characteristics are initialized from VS034 array. Laminar pressure drop for leakage from pressure to return is calculated as

$$\Delta P = CKLxQ$$

and turbulent pressure drop for normal flow through the valve is calculated as

$$\Delta P = CKTxQ^2$$

Where ΔP -	Pressure drop in psi
Q -	flow in gpm
CKL -	laminar flow coefficient
CKT -	turbulent flow coefficient

6.2.12.4 Approximations - Not applicable

6.2.12.5 Limitations - Not applicable

6.2.12.6 DVS034 Variable Names

Variable	Description	Dimension
ALT	Altitude	FT
ARRAY1	Computational array	--
BLEG	General purpose array	--
BRANCHP	Dynamic element data storage array	--
CK	Resistance coefficient	--
CKL	Laminar leakage coefficient	PSI/GPM
CKT	Turbulent leakage coefficient	PSI/GPM
DENS	Fluid weight density at atmospheric pressure	LB/FT^3
DPL1, DPL2, DPL3, DPL4, DPL5	Resistance factor for valve internal leg	PSI/GPM^2
DP1	Minimum pressure at rated flow	PSI
DP2	Pressure difference between inlet and outlet	PSI
D100	Weight density at 100°F	LB/FT^3
FLUIDF	Viscosity-pressure correction factor at 100°F	--
FLUIDK	Viscosity-pressure correction factor at fluid temperature	--
IL	Row location in BRANCHP array	--
IND	Indicator to FLOCHEK subroutine	--

Variable	Description	Dimension
IP1	Upstream pressure point number for flow regulator type 37	--
IP2	Downstream pressure point number for flow regulator type 37	--
J	Integer counter	--
K	Row location in BRANCHP	--
LEG	Leg number	--
M	Operating code (see Figure 4-9)	--
MLEG	Dummy array for leg numbers	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG array	--
NPQ	Total number of rows in PQL array	--
PAMB	Atmospheric ambient pressure	--
PQL	Array containing pressure point data	--
P1	Inlet pressure	PSI
P2	Outlet pressure	PSI
RPL	Rated pressure drop for leakage conditions	PSI
RPT	Rated pressure drop for rated flow	PSI
RQL	Rated flow of flow regulator	GPM
RQT	Rated flow of valve	GPM
RQ1	Rated flow for flow regulator type 37	GPM
TEMP	Fluid temperature	°F
TY	Element type	--
VISC	Fluid viscosity at atmospheric pressure	CENTISTOKES
VRATIO	Viscosity ratio	--
V100	Fluid viscosity at 100°F	CENTISTOKES

6.2.12.7 DVS034 - Subroutine Listing

```
      SUBROUTINE DVS034(TY,IL,BRANCHP,NBP2,BLEG,NL,PQL,NPQ)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION BRANCHP(1),BLEG(1),PQL(1)
      DIMENSION MLEG(5),ARRAY1(5,3)
      K=IL
      IF(TY.EQ.37..OR.TY.EQ.38.)GO TO 10
      MLEG(1)=BRANCHP(K+15*NBP2)
      MLEG(2)=BRANCHP(K+16*NBP2)
      MLEG(3)=BRANCHP(K+17*NBP2)
      MLEG(4)=BRANCHP(K+18*NBP2)
      MLEG(5)=BRANCHP(K+19*NBP2)
      M=BRANCHP(K+14*NBP2)
      RPT=BRANCHP(K+10*NBP2)
      RQT=BRANCHP(K+9*NBP2)
      RPL=BRANCHP(K+13*NBP2)
      IF(BRANCHP(K+12*NBP2).EQ.0.)BRANCHP(K+12*NBP2)=.0001
      RQL=BRANCHP(K+12*NBP2)
      CKT=RPT/RQT**2.0
      CKL=RPL/RQL
      VRATIO=(VISC/BRANCHP(K+11*NBP2))
      IF(TY.NE.36.)GO TO 2
C    *****TYPE#36 TWO WAY-TWO POSITION SOLENOID VALVE*****
      ARRAY1(1,1)=ABS(BLEG(MLEG(1)+2*NL))
      ARRAY1(1,2)=CKT*(ARRAY1(1,1))**2.0
      ARRAY1(1,2)=(ARRAY1(1,2)*VRATIO**0.25)/ARRAY1(1,1)
      ARRAY1(1,3)=CKL*ARRAY1(1,1)
      ARRAY1(1,3)=(ARRAY1(1,3)*VRATIO)/ARRAY1(1,1)
      IF(M.NE.1) GO TO 1
C    *****VALVE IS OPEN AND FLOW IS TURBULENT*****
      DPL1=ARRAY1(1,2)
      BLEG(MLEG(1)+11*NL)=BLEG(MLEG(1)+11*NL)+DPL1
      GO TO 9
    1 CONTINUE
C    *****VALVE IS CLOSED AND FLOW IS LAMINAR*****
      DPL1=ARRAY1(1,3)
      BLEG(MLEG(1)+11*NL)=BLEG(MLEG(1)+11*NL)+DPL1
      GO TO 9
    2 CONTINUE
      IF(TY.NE.35.)GO TO 5
C    *****TYPE#35 THREE WAY-TWO POSITION SOLENOID VALVE*****
      DO 3 J=1,3
      ARRAY1(J,1)=ABS(BLEG(MLEG(J)+2*NL))
      ARRAY1(J,2)=CKT*(ARRAY1(J,1))**2.0
      ARRAY1(J,2)=(ARRAY1(J,2)*VRATIO**.25)/ARRAY1(J,1)
      ARRAY1(J,3)=CKL*ARRAY1(J,1)
      ARRAY1(J,3)=(ARRAY1(J,3)*VRATIO)/ARRAY1(J,1)
    3 CONTINUE
      IF(M.NE.1) GO TO 4
C    *****VALVE IS OPEN P-C FLOW IS TURBULENT*****
      DPL1=ARRAY1(1,3)
```

6.2.12.7 (Continued)

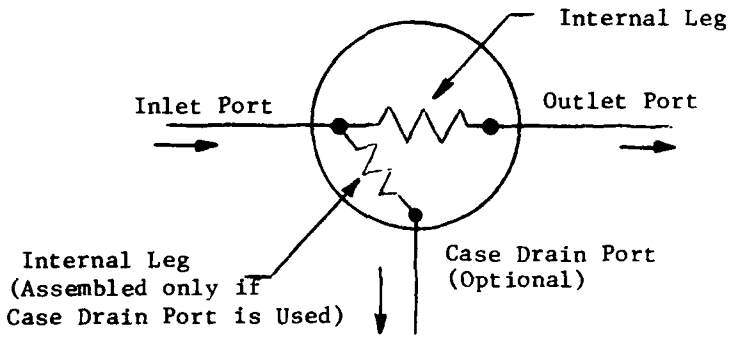
```
      DPL2=ARRAY1(2,2)
      BLEG(MLEG(1)+11*NL)=BLEG(MLEG(1)+11*NL)+DPL1
      BLEG(MLEG(2)+11*NL)=BLEG(MLEG(2)+11*NL)+DPL2
      BLEG(MLEG(3)+11*NL)=3.E07
      GO TO 9
    4 CONTINUE
C   *****VALVE IS CLOSED C-R FLOW IS TURBULENT*****
      DPL1=ARRAY1(1,3)
      DPL3=ARRAY1(3,2)
      BLEG(MLEG(1)+11*NL)=BLEG(MLEG(1)+11*NL)+DPL1
      BLEG(MLEG(2)+11*NL)=3.E07
      BLEG(MLEG(3)+11*NL)=BLEG(MLEG(3)+11*NL)+DPL3
      GO TO 9
    5 CONTINUE
      IF(TY.NE.34.)GO TO 9
C   *****TYPE#34 FOUR WAY-THREE POSITION SOLENOID VALVE*****
      DO 6 J=1,5
      ARRAY1(J,1)=ABS(BLEG(MLEG(J)+2*NL))
      ARRAY1(J,2)=CKT*(ARRAY1(J,1)**2.0)
      ARRAY1(J,2)=(ARRAY1(J,2)*VRATIO**.25)/ARRAY1(J,1)
      ARRAY1(J,3)=CKL*ARRAY1(J,1)
      ARRAY1(J,3)=(ARRAY1(J,3)*VRATIO)/ARRAY1(J,1)
    6 CONTINUE
      IF(M.EQ.2) GO TO 7
      IF(M.EQ.3) GO TO 8
C    ***** VALVE IS OPEN P-C3  AND  C4-R *****
      DPL1=ARRAY1(1,3)
      DPL2=ARRAY1(2,2)
      DPL5=ARRAY1(5,2)
      BLEG(MLEG(1)+11*NL)=BLEG(MLEG(1)+11*NL)+DPL1
      BLEG(MLEG(2)+11*NL)=BLEG(MLEG(2)+11*NL)+DPL2
      BLEG(MLEG(3)+11*NL)=3.E07
      BLEG(MLEG(4)+11*NL)=3.E07
      BLEG(MLEG(5)+11*NL)=BLEG(MLEG(5)+11*NL)+DPL5
      GO TO 9
    7 CONTINUE
C
C    ***** VALVE IS OPEN P-C4  AND C3-R *****
C
      DPL1=ARRAY1(1,3)
      DPL3=ARRAY1(3,2)
      DPL4=ARRAY1(4,2)
      BLEG(MLEG(1)+11*NL)=BLEG(MLEG(1)+11*NL)+DPL1
      BLEG(MLEG(2)+11*NL)=3.E07
      BLEG(MLEG(3)+11*NL)=BLEG(MLEG(3)+11*NL)+DPL3
      BLEG(MLEG(4)+11*NL)=BLEG(MLEG(4)+11*NL)+DPL4
      BLEG(MLEG(5)+11*NL)=3.E07
      GO TO 9
    8 CONTINUE
C
```

6.2.12.7 (Continued)

```
C     ***** VALVE IS CLOSED *****
C
      DPL1=ARRAY1(1,3)
      BLEG(MLEG(1)+11*NL)=BLEG(MLEG(1)+11*NL)+DPL1
      BLEG(MLEG(2)+11*NL)=3.E07
      BLEG(MLEG(3)+11*NL)=3.E07
      BLEG(MLEG(4)+11*NL)=3.E07
      BLEG(MLEG(5)+11*NL)=3.E07
    9 CONTINUE
      RETURN
   10 IND=BRANCHP(K+8*NBP2)
      RQ1=BRANCHP(K+5*NBP2)
      IP1=BRANCHP(K+3*NBP2)
      IP2=BRANCHP(K+4*NBP2)
      LEG=BRANCHP(K+15*NBP2)
      IF(IND.GE.1)GO TO 11
      PQL(IP1+NPQ)=(-RQ1)
      PQL(IP2+NPQ)=RQ1
      BLEG(LEG+11*NL)=3.E9
      RETURN
   11 P1=PQL(IP1)
      P2=PQL(IP2)
      DP1=BRANCHP(K+6*NBP2)
      DP2=P1-P2
      IF(DP1.LE.0.)DP1=1.
      CK=DP1/RQ1
      BLEG(LEG+11*NL)=CK
      RETURN
      END
```

6.2.13 Subroutine DMTR8

DMTR8 is a dynamic subroutine for calculating the internal flows and pressure losses in a hydraulic motor. The motor may be a 2 port or 3 port model, see Figure 53.


Internal Leg
Inlet Port
Outlet Port
Internal Leg (Assembled only if Case Drain Port is Used)
Case Drain Port (Optional)

Hydraulic Motor
FIGURE 53

6.2.13.1 Math Model

The motor losses are calculated and input into BLEG column 5. If the motor should be stalled, MTRCHK places an indicator in the data column 15. If there is an indicator in this position, a high resistance is placed in BLEG column 12.

6.2.13.2 Assumptions - A minimum break out torque of 60 psi is assumed.

6.2.13.3 Computations - Input Torque = Output Torque (Load)/Efficiency

$$\text{PRESS DROP} = 2\pi * \text{INPUT TORQUE/DISPLACEMENT}$$

6.2.13.4 Approximations - Not applicable.

6.2.13.5 Limitations - Not applicable.

6.2.13.6 DMTR8 Variable Names

Variable	Description	Dimension
BLEG	General purpose array	--
BRANCHP	Dynamic element data storage array	--
CDK	Leakage coefficient	--
CDL	Rated case drain leakage flow	GPM/3000 PSID
DELP	Pressure difference between inlet and outlet	PSID
DISPL	Motor displacement	IN^3/REV
DP	Pressure drop	PSID
EFF	Overall motor efficiency	%
IL	Element row location in BRANCHP	--
IND	Indicates normal or stalled condition 0=normal 1=stalled	--
LEG	Leg number inlet to outlet	--
LEG1	Leg number inlet to case	--
NBP2	Total number of rows in BRANCHP array	--
NDP	Case drain pressure point number (If used)	--
NL	Total number of rows in BLEG and ILEP arrays	--
NPQ	Total number of rows in PQL array	--
PQL	Array of pressure points, flows and port numbers	--
PR1	Inlet pressure	PSI
PR2	Outlet pressure	PSI
P1	Inlet pressure point number	--
P2	Outlet pressure point number	--
Q	Flow rate	GPM

Variable	Description	Dimension
TORQL	Load torque	IN-LB
TT	Equivalent inlet torque	IN-LB

6.2.13.7 DMTR8 Subroutine Listing

```
      SUBROUTINE DMTR8(IL,BRANCHP,NBP2,PQL,NPQ,BLEG,NL)
      DIMENSION BRANCHP(1),PQL(1),BLEG(1)
      INTEGER P1,P2
      LEG=BRANCHP(IL+15*NBP2)
      LEG1=BRANCHP(IL+16*NBP2)
      IND=BRANCHP(IL+14*NBP2)
      Q=BLEG(LEG+2*NL)
      BRANCHP(IL+13*NBP2)=Q
      P1=BRANCHP(IL+4*NBP2)
      P2=BRANCHP(IL+6*NBP2)
      DISPL=BRANCHP(IL+7*NBP2)
      TORQL=BRANCHP(IL+8*NBP2)
      EFF=BRANCHP(IL+9*NBP2)
      CDL=BRANCHP(IL+10*NBP2)
      IF(IND.GE.1)GO TO 1
      TT=TORQL/EFF
      DP=2.*3.14159*TT/DISPL
      BLEG(LEG+4*NL)=-DP-60.
      GO TO 2
    1 BLEG(LEG+11*NL)=3.E10
    2 NDP=BRANCHP(IL+6*NBP2)
      IF(NDP.EQ.0)RETURN
      PR1=PQL(P1)
      PR2=PQL(P2)
      DELP=PR1-PR2
      CDK=3000./CDL
      BLEG(LEG1+11*NL)=CDK
      RETURN
      END
```

SECTION VII
OUTPUT

The output section of SSFAN is contained in the main program and special purpose subroutines. The input data cards are printed from the main program immediately after they are read and before any data processing is done. OPUT4 subroutine prints the heading and data deck title. OPUT3 subroutine prints the system assembly of legs and pressure points and OPUT2 subroutine prints the corresponding pressures and flow rates. QTCALC subroutine prints the quasi-transient data in conjunction with GRAPH2 subroutine.

7.1 OPUT4 Subroutine

The heading and data deck title are printed from OPUT4, see Figure 54. This output is used for the input data card output, the assembly output and the calculated pressure and flow output.

7.2 OPUT3 Subroutine

The system assembly output is printed from OPUT3 subroutine, see Figure 55. Leg numbers and branch point numbers are computer assigned and are cross referenced to junction numbers in this output. The leg number is printed with its upstream and downstream pressure point number. The corresponding upstream and downstream junction numbers are next printed.

The elements that have assigned pressure point numbers are next output with its name and junction number(s).

7.3 OPUT2 Subroutine

OPUT2 subroutine is used to print the flow rates and pressures, Figure 56, in a corresponding order to the assembly output from subroutine OPUT3. The computer assigned leg number and calculated flow rate for the leg are first printed. Next, the computer assigned pressure point number and the calculated

```
********************************************************************************
**********************************SSFAN PROGRAM*********************************
********************************************************************************

DATA DECK---------------SSFAN  --  SAMPLE CASE NUMBER 1

*********************************COLUMN NUMBERS*********************************
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
++++++++$$$$$$$$++++++++$$$$$$$$++++++++$$$$$$$$++++++++$$$$$$$$++++++++$$$$$$$$

CARD 1 +++++++++++++++++++++++TITLE++++++++++++++++++++++++++++++++++++++++++++++
SSFAN  --  SAMPLE CASE NUMBER 1

CARD 2 ++++++++++++++++++++++FLUID+++++++++++++++++++++++++++++++++++++++++++++++
MIL-H-5606A

CARDS 3&4 +++++++++VISCOSITY-TEMPERATURE DATA++++++++++++++++++++++++++++++++++++
9       2000.00   385.00   130.00    35.00    14.50     7.70     4.80     3.40     2.40
9        -55.00   -40.00     0.00    50.00   100.00   150.00   200.00   250.00   300.00

CARD 5 +++++++++++++DENSITY-TEMPERATURE DATA+++++++++++++++++++++++++++++++++++++
  56.200  49.900 -65.000 275.000

CARD 6 +++++INITIAL TEMP-FINAL TEMP-TEMP INCR-ALTITUDE-PRINT-OPTIONS+++++++++++++
 100.000 100.000   0.000   0.000   1.000
```

FIGURE 54

OUTPUT - SSFAN TITLE AND DATA CARDS 1-6

```
LEG  PRESSURE PT   BRANCH/END PT * PRESS  ELEMENT          JUNCTION
NO.  (UP)   (DN)    (UP)    (DN) * PT NO                    NUMBERS
 1     2     15     10.      35. *    1    PUMP               5.
 2     1     11      5.     265. *    2    PUMP              10.
 3     3     12     15.     215. *    3    PUMP              15.
 4     9     12    250.     230. *    4    ACCUM            295.
 5    10     15    260.      45. *    5    ACTR             145.
 6    12     18    210.      90. *    6    ACTR             185.
 7    15     14     40.      55. *    7    ACTR             130.
 8    14      4     65.     295. *    8    ACTR             170.
 9    14     17     60.      85. *    9    RESV             250.
10    19     13     95.     115. *   10    RESV             260.
11    13      7    120.     130. *   11    RESV             265.
12     8     16    170.     160. *   12    TEE         210.- 215.- 230.
13    13      5    135.     145. *   13    TEE         115.- 120.- 135.
14     6     16    185.     175. *   14    TEE          55.-  65.-  60.
15    16     20    155.     100. *   15    TEE          35.-  45.-  40.
16     5      6    145.     185. *   16    TEE         175.- 160.- 155
17     7      8    130.     170. *   17    VLV               85.
18    17     18     85.      90. *   18    VLV               90.
19    17     19     85.      95. *   19    VLV               95.
20    19     18     95.      90. *   20    VLV              100.
21    17     20     85.     100. *
22    20     18    100.      90. *
```

FIGURE 55

DATA FROM OPUT3 SUBROUTINE

```
LEG      FLOW     **   PRESSURE     PRESSURE    TEMPERATURE
NO.      (GPM)    **   PT NO.       (PSIG)       (DEG F)
  1      12.84    **      1           50.56       100.00
  2     -13.50    **      2         3075.38       100.00
  3        .66    **      3          135.85       100.00
  4      -9.86    **      4         2906.82       100.00
  5       -.41    **      5          976.46       100.00
  6      -9.20    **      6         1044.30       100.00
  7      12.44    **      7         1155.98       100.00
  8        .00    **      8         1189.82       100.00
  9      12.44    **      9           53.07       100.00
 10      12.36    **     10         3056.95       100.00
 11       6.45    **     11           53.07       100.00
 12       4.86    **     12          130.66       100.00
 13       5.91    **     13         1294.73       100.00
 14       4.27    **     14         2906.82       100.00
 15       9.13    **     15         3063.18       100.00
 16       5.91    **     16          846.30       100.00
 17       6.45    **     17         2499.28       100.00
 18        .07    **     18          206.08       100.00
 19      12.36    **     19         2449.85       100.00
 20        .00    **     20          233.03       100.00
 21        .00    **
 22       9.13    **
```

FIGURE 56

DATA FROM OPUT2 SUBROUTINE

```
CARDS 7 & ON ++++++++++++++++++++++++++++++ELEMENT DATA++++++++++++++++++++++++++++++
************************************COLUMN NUMBERS**********************************
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
++++++++++$$$$$$$$$$++++++++++$$$$$$$$$$++++++++++$$$$$$$$$$++++++++++$$$$$$$$$$
2             4.000   145.000   155.000     4.000     4.000     4.500     3.250    35.0001000.000
     5.000     2.500     2.000    95.000     0.000     0.000     0.000     0.000     0.000     0.000
2             4.000   130.000   170.000     4.000     4.000     3.850     2.900    25.0001000.000
     5.000     2.500     2.000    95.000     0.000     0.000     0.000     0.000     0.000     0.000
2             1.000    40.000    50.000     4.000     .0065   142.500    30.000    45.000   120.000
    65.000    23.000    72.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
1             1.000   230.000   235.000    12.000      .028    63.200    42.000    33.200     0.000
2            34.000    45.000    90.000    95.000   100.000     6.000     8.000     4.000     4.000
    25.000   200.000    14.000      .1003000.000     1.000     0.000     0.000     0.000     0.000
2             0.000   250.000   260.000   265.000    10.000     4.000    16.000     1.1655    95.300
     2.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
2             1.000   205.000   210.000     4.000      .023   147.300    23.000    38.000    90.000
    63.000   121.000   150.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
2             5.000     5.000    10.000    15.000    16.000    12.000     4.0003750.0003750.000
    50.0003000.0002750.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
1             2.000   245.000   250.000    10.000    10.000     0.000     0.000     0.000     0.000
1             2.000   235.000   240.000    10.000    10.000     0.000     0.000     0.000     0.000
1             2.000   220.000   215.000     4.000     4.000     0.000     0.000     0.000     0.000
1             1.000   225.000   220.000     4.000      .020    78.200    97.000    93.000    93.000
1                          .000   225.000     4.000     4.000              0.000
1                                 205.000      .000

1
1             2.000   275.000   230.000    1          15.000     0.000     0.000     0.000     0.000
1            21.000    25.000    30.000    12.000    12.000     0.000     0.000     0.000     0.000
1             2.000   285.000     5.000    16.000    15.000     0.000     0.000     0.000     0.000
1             7.000   272.000     4.000     1.000    50.000   100.000     4.000     6.000     0.000
1             0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
```

FIGURE 57
EXAMPLE SSFAN SAMPLE CASE NUMBER 1
ELEMENT INPUT DATA - CARDS 7 AND ON

pressure are printed. The input temperature for the pressure point is printed in the last column.

7.4 Main Program Output Data

The main program prints out the data cards immediately after they are read. This data may be used for trouble shooting of incorrect input data. The column number headings are printed at the top of the data with data fields of column width 8, identified by alternating +'s and $'s. Data cards 1 through 6, system parameters are shown in Figure 54. Figure 57 shows the input data from cards 7 and on, immediately after it is read.

7.5 Quasi - Transient Data Output

The quasi-transient data output is of 3 types; (1) Pressure versus time, (2) Flow versus time and (3) Piston position versus time. These are user selected and an example of each type is shown in Figures 58, 59, 60 and 61. The type of output is printed at the bottom of each graph along with the system data deck title. The junction number associated with the output is also printed. The tabulated data of Figure 58 contains all the calculated time steps and may contain more points than are plotted on the graphs.

```
                 SSFAN  --  SAMPLE CASE NUMBER 1

ROW 2 END  1=PRESS PLOT  2=FLOW PLOT  3=PIST POS PLOT


       TIME        JCT         JCT         JCT
                 145.000     145.000     145.000
                   2.000       1.000       3.000
                   0.000       0.000       1.000
                   0.000       0.000       0.000
                   0.000       0.000       5.000
                   0.000       0.000       0.000
     0.0000        0.000    1540.270       0.000
      .0500        0.000    1540.274       0.000
      .1000        0.000    1540.274       0.000
      .1500        0.000    1540.274       0.000
      .2000        0.000    1540.274       0.000
      .2500        0.000    1540.274       0.000
      .3000        0.000    1540.274       0.000
      .3500        0.000    1540.274       0.000
      .4000        0.000    1540.274       0.000
      .4500        0.000    1540.274       0.000
      .5000        0.000    1540.274       0.000
      .5500        2.155    2292.819        .092
      .6000        2.344    2309.449        .193
      .6500        2.443    2321.160        .297
      .7000        2.533    2332.894        .405
      .7500        2.613    2344.646        .517
      .8000        2.683    2356.582        .632
      .8500        2.744    2368.680        .749
      .9000        2.796    2380.915        .869
      .9500        2.840    2393.267        .990
     1.0000        2.876    2405.717       1.113
     1.0500        2.906    2418.246       1.238
     1.1000        2.929    2430.835       1.363
     1.1500        2.947    2443.467       1.489
     1.2000        2.960    2456.125       1.616
     1.2500        2.967    2468.793       1.743
     1.3000        2.970    2481.516       1.870
     1.3500        2.968    2494.230       1.997
     1.4000        2.963    2506.909       2.123
     1.4500        2.954    2519.540       2.250
     1.5000        2.941    2532.111       2.376
     1.5500        2.926    2544.610       2.501
     1.6000        2.907    2557.027       2.625
     1.6500        2.882    2569.347       2.748
     1.7000        2.856    2581.546       2.871
     1.7500        2.829    2593.625       2.992
     1.8000        2.800    2605.580       3.111
     1.8500        2.768    2617.403       3.230
     1.9000        2.736    2629.084       3.347
     1.9500        2.704    2640.622       3.463
     2.0000        2.670    2652.017       3.577
     2.0500        2.637    2663.244       3.690
     2.1000        2.603    2674.290       3.801
     2.1500        2.568    2685.212       3.911
     2.2000        2.532    2695.946       4.019
     2.2500        2.496    2706.526       4.126
     2.3000        3.454    2713.894       4.274
     2.3500        3.358    2728.575       4.417
     2.4000        3.265    2742.933       4.557
     2.4500        3.174    2756.993       4.693
     2.5000        3.084    2770.638       4.825
     2.5500        2.993    2783.844       4.953
     2.6000        2.903    2796.479       5.000
     2.6500         .092    3009.956       5.000
     2.7000         .092    3009.956       5.000
     2.7500         .092    3009.956
```

FIGURE 58 QUASI-TRANSIENT TABULATED DATA

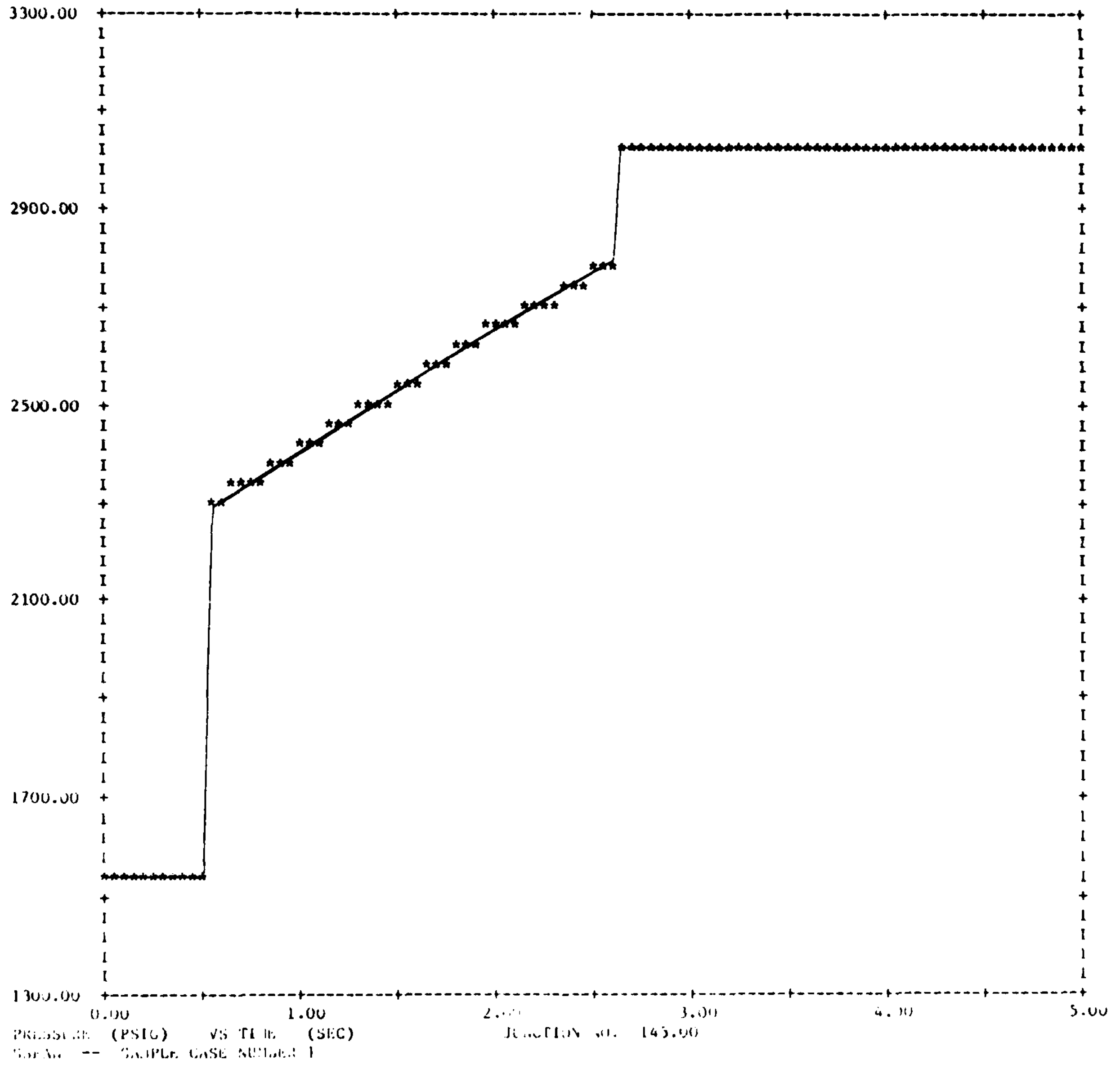


FIGURE 59 QUASI-TRANSIENT PRESSURE
VERSUS
TIME COMPUTER PLOT

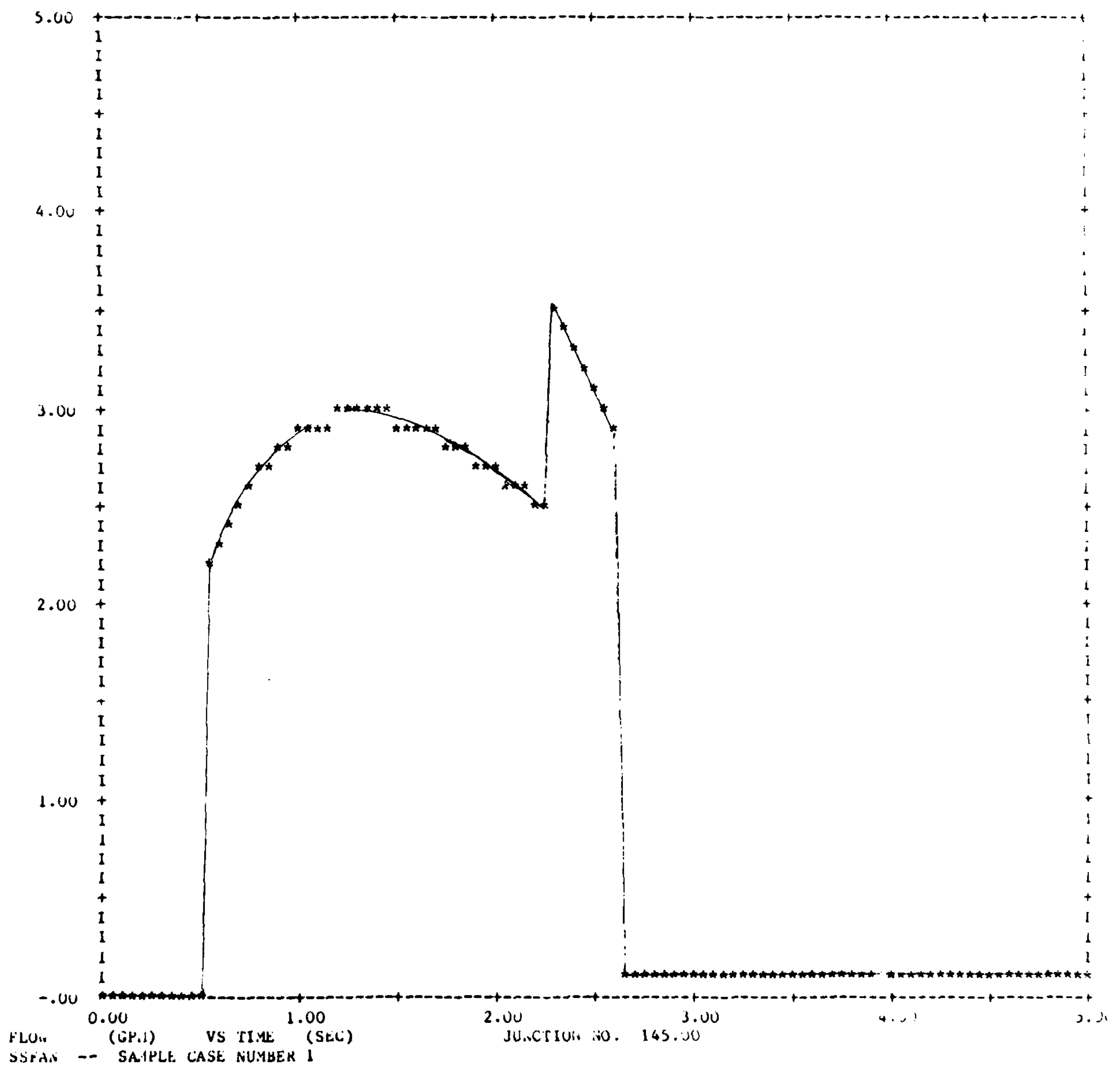


FIGURE 60 QUASI-TRANSIENT FLOW
VERSUS
TIME COMPUTER PLOT

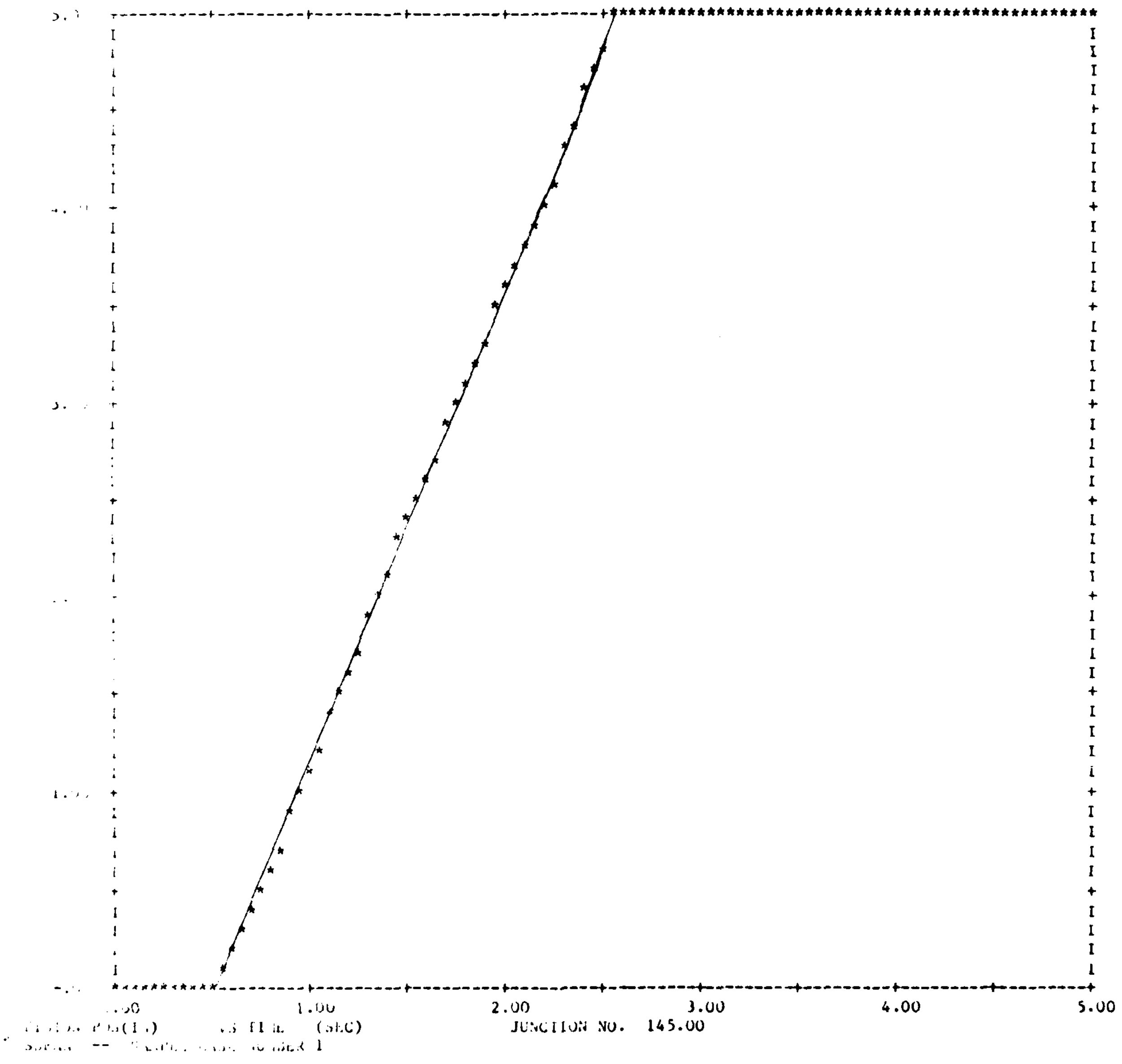


FIGURE 61 QUASI-TRANSIENT ACTUATOR POSITION
VERSUS
TIME COMPUTER PLOT

7.6 OPUT4 Subroutine Listing

```
  SUBROUTINE OPUT4
  COMMON /BLK9/FTYPE(16),PRINT(4)
  WRITE(6,3)
1 FORMAT(' ',80('*'))
  WRITE(6,2)
  WRITE(6,1)
2 FORMAT(' ',34('*'),'SSFAN PROGRAM',33('*'))
3 FORMAT('1',80('*'))
  WRITE(6,4)(FTYPE(M),M=1,8)
4 FORMAT(//' ','DATA DECK---------------',8A10//)
  RETURN
  END
```

7.7 OPUT3 Subroutine Listing

```
      SUBROUTINE OPUT3(ILEP,BLEG,NL,PQL,NPQ)
C        OPUT3 LISTS LEG NUMBERS UP & DN STREAM PRESS PTS & JCT NO
      DIMENSION ILEP(1),BLEG(1),PQL(1)
      COMMON /BLK9/FTYPE(16),PRINT(4)
      WRITE(108,1)(FTYPE(M),M=1,8)
    1 FORMAT(8A10)
      WRITE(108,2)
    2 FORMAT(//59HLEG NO-LEG END PRESSURE PT NO-JUNCTION PT NO & ELEMENT
     1 TYPE)
      WRITE(108,3)
    3 FORMAT(//,1X,3HLEG,2X,11HPRESSURE PT,3X,13HBRANCH/END PT,1X,1H*,1X
     1,5HPRESS,2X,7HELEMENT,9X,8HJUNCTION)
      WRITE(108,4)
    4 FORMAT(1X,3HNO. 2X,4H(UP),3X,4H(DN),4X,4H(UP),4X,4H(DN),1X,1H*,1X,
     15HPT NO,19X,7HNUMBERS)
      DO 25 M=1,100
      IF(ILEP(M).EQ.0.AND.PQL(M+2*NPQ).EQ.0.)GO TO 26
      IF(PQL(M+2*NPQ).EQ.0.)GO TO 21
      IF(PQL(M+2*NPQ).EQ.5.)GO TO 7
      IF(PQL(M+2*NPQ).EQ.7.)GO TO 9
      IF(PQL(M+2*NPQ).EQ.9.)GO TO 11
      IF(PQL(M+2*NPQ).EQ.91.)GO TO 11
      IF(PQL(M+2*NPQ).EQ.92.)GO TO 11
      IF(PQL(M+2*NPQ).EQ.4..OR.PQL(M+2*NPQ).EQ 41.)GO TO 13
      IF(PQL(M+2*NPQ).EQ.24.)GO TO 15
      IF(PQL(M+2*NPQ).EQ.25.)GO TO 17
      IF(PQL(M+2*NPQ).EQ.13.)GO TO 5
      IF(PQL(M+2*NPQ).EQ.34.)GO TO 19
      IF(PQL(M+2*NPQ).EQ.8.)GO TO 23
      IF(PQL(M+2*NPQ).EQ.35.)GO TO 19
      IF(PQL(M+2*NPQ).EQ.36.)GO TO 19
      IF(PQL(M+2*NPQ) EQ.37.)GO TO 19
      IF(PQL(M+2*NPQ).EQ 38.)GO TO 19
      GO TO 25
    5 WRITE(108,6)M,ILEP(M),ILEP(M+NL),BLEG(M),BLEG(M+NL),M,PQL(M+3*NPQ)
    6 FORMAT(I3,I6,I7,F9.0 F8.0,1X,1H*,1X,I5,4X,3HFBP,F17.0)
      GO TO 25
    7 WRITE(108,8)M,ILEP(M),ILEP(M+NL),BLEG(M),BLEG(M+NL),M PQL(M+3*NPQ)
    8 FORMAT(I3,I6,I7,F9.0,F8.0,1X,1H*,1X,I5,4X,4HPUMP,F16.0)
      GO TO 25
    9 WRITE(108,10)M,ILEP(M),ILEP(M+NL),BLEG(M),BLEG(M+NL),M,PQL(M+3*NPQ
     1)
   10 FORMAT(I3,I6,I7,F9.0,F8.0,1X,1H*,1X,I5,4X,5HACCUM,F15.0)
      GO TO 25
   11 WRITE(108,12)M,ILEP(M),ILEP(M+NL),BLEG(M),BLEG(M+NL),M,PQL(M+3*NPQ
     1)
   12 FORMAT(I3,I6,I7,F9.0,F8.0,1X,1H*,1X,I5,4X,4HRESV,F16.0)
      GO TO 25
   13 WRITE(108,14)M,ILEP(M),ILEP(M+NL),BLEG(M) BLEG(M+NL),M,PQL(M+3*NPQ
     1)
```

```
14 FORMAT(I3,I6,I7.F9.0,F8.0 1X,1H*,1X,I5,4X,4HACTR,F16.0)
   GO TO 25
15 WRITE(108,16)M,ILEP(M),ILEP(M+NL),BLEG(M),BLEG(M+NL),M,PQL(M+3*NPQ
  1),PQL(M+4*NPQ),PQL(M+5*NPQ)
16 FORMAT(I3,I6,I7,F9.0,F8.0 1X,1H*,1X I5,4X,3HTEE,F11.0,1H-,F5.0,1H-
  1,F5.0)
   GO TO 25
17 WRITE(108,18)M,ILEP(M),ILEP(M+NL),BLEG(M),BLEG(M+NL),M,PQL(M+3*NPQ
  1)
  2PQL(M+4*NPQ),PQL(M+5*NPQ),PQL(M+6*NPQ)
18 FORMAT(I3,I6,I7,F9.0,F8.0,1X,1H*,1X,I5,4X,5HCROSS,F8.0,1H-,F5.0,1
  1H-,F5.0,1H-,F5.0)
   GO TO 25
19 WRITE(108,20)M,ILEP(M),ILEP(M+NL),BLEG(M),BLEG(M+NL),M,PQL(M+3*NPQ
  1)
20 FORMAT(I3,I6,I7,F9.0,F8.0,1X,1H*,1X,I5,4X,3HVLV,F18.0)
   GO TO 25
21 WRITE(108,22)M,ILEP(M),ILEP(M+NL),BLEG(M),BLEG(M+NL)
22 FORMAT(I3,I6,I7,F9.0,F8.0,1X,1H*)
   GO TO 25
23 WRITE(108,24)M,ILEP(M),ILEP(M+NL),BLEG(M),BLEG(M+NL),M,PQL(M+3*NPQ
  1)
24 FORMAT(I3,I6,I7,F9.0,F8.0,1X,1H*,1X,I5,4X.5HMOTOR,F15.0)
25 CONTINUE
26 RETURN
   END
```

7.8 OPUT2 Subroutine Listing

```
      SUBROUTINE OPUT2(ILEP,BLEG,NL,PQL,NPQ)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DIMENSION ILEP(1),BLEG(1),PQL(1)
      COMMON /BLK7/IERROR,ITER
      WRITE(108,1)ITER
    1 FORMAT(//13HITERATIONS = ,I3)
      WRITE(108,2)ALT
    2 FORMAT(11HALTITUDE = F8.2,1X,2HFT)
      WRITE(108,3)PAMB
    3 FORMAT(19HAMBIENT PRESSURE = ,F8.2,1X,4HPSIA)
      WRITE(108,4)
    4 FORMAT(///3HLEG,6X,4HFLOW,5X,2H**,5X,8HPRESSURE,6X,8HPRESSURE,6X,1
     11HTEMPERATURE,/3HNO.,6X,5H(GPM),4X,2H**,6X,6HPT NO.,8X,6H(PSIG),8X
     2,7H(DEG F))
      DO 10 M=1,10000
      IF(ILEP(M).EQ.0.AND.PQL(M).EQ.0.)GO TO 11
      IF(ILEP(M).EQ 0.)GO TO 8
      IF(PQL(M).EQ.0.)GO TO 6
      WRITE(108,5)M,BLEG(M+2*NL) M,PQL(M),BLEG(M+13*NL)
    5 FORMAT(I3,2X,F9.2,4X,2H**,6X,I5,6X,F10.2,8X,F7.2)
      GO TO 10
    6 WRITE(108,7)M,BLEG(M+2*NL)
    7 FORMAT(I3,2X,F9.2,4X,2H**)
      GO TO 10
    8 WRITE(108,9)M PQL(M),BLEG(M+13*NL)
    9 FORMAT(18X,2H**,6X,I5,6X,F10.2,8X,F7.2)
   10 CONTINUE
   11 RETURN
      END
```

SECTION VIII

UTILITY SUBROUTINES

8.1 INTERP Subroutine

The INTERP subroutine provides interpolation for continuous or discontinuous functions of the form Y = f(X). INTERP is a shortened version of a MCAUTO library functional subroutine named DISCOT.

INTERP uses two other subroutines, DISER1 and LAGRAN, to derive the dependent variable from tabulated data input by the programmer. Subroutine DISER1 gives the data points around the X variable. Lagranges interpolation formula is used in the LAGRAN subroutine to obtain a Y value. For an X value lying outside the range of the tabulated data, the Y is extrapolated. Fluid viscosities are calculated in LAGRAN by using the ASTM equation for viscosity (See Appendix B). A fluid viscosity - pressure correction factor FLUIDF is also calculated from the ASTM viscosity/temperature slope and slope/constant equations as described in Appendix B.

8.1.1 Solution Method

The INTERP subroutine provides the necessary control parameters to DISER1 and LAGRAN to yield a dependent variable. The subroutine arguments are as follows:

Subroutine INTERP (X, TABX, TABY, NC, NY, Y, IND)

Where:

X	- Argument of function Y = f(X)
TABX	- X array of independent variables in ascending order
TABY	- Y array of dependent variables
NC	- Control word

Tens Digit	-	Degree of interpolation
Units Digit	-	0 = Lagrange interpolation
		1 = Viscosity calculation with ASTM equation
		2 = FLUIDF calculation

NY	- Number of data points in the Y array
Y	- Dependent variable
IND	- Interpolation Indicator
	0 = Normal interpolation
	1 = Extrapolation outside the range of data points

8.1.2 Assumptions. Not applicable

8.1.3 Computations. The degree of interpolation is decoded from the control word NC in the INTERP subroutine argument and passed to DISER1. The error indicator IND is set to zero. On finding the data point closest to the X value from DISER1, it is entered into the LAGRAN subroutine argument. If the ASTM equation is to be used for a viscosity calculation, IDX is set to -1. For the FLUIDF computation IDX equals -2.

8.1.4 Approximations. Not applicable

8.1.5 Limitations. The X data points must be entered in an ascending order. When tabulating a discontinuous function the independent variable (X) at the point of discontinuity is repeated, i.e.,

$$X_1, X_2, X_3, X_3, X_4, X_5$$
$$Y_1, Y_2, Y_3, Y_4, Y_5, Y_6$$

Thus for discontinuous functions there must be K + 1 points above and below the discontinuity, where K is the degree of interpolation.

8.1.6 INTERP Variable Names

Variables	Description	Dimensions
IDX	Degree of interpolation	-
IND	Solution indicator	-
	= 0 Normal interpolation	
	= 1 Extrapolation outside of data range	
NC	Control word	-
NPX	Dummy array	-
NPX1	Location of data point X, Y for interpolation	-
NY	Number of Y data points	-
TABX	X array of data points	-
TABY	Y array of data points	-
X, XA	Independent variable	-
Y	Dependent variable	-

8.1.7 <u>INTERP Subroutine Listing</u>

```
SUBROUTINE INTERP(X,TABX,TABY,NC,NY,Y,IND)
DIMENSION TABX(1),TABY(1),NPX(8)
IDX=(NC-(NC/100)*100)/10
IND=0
XA=X
CALL DISER1(XA,TABX,1,NY,IDX,NPX,IND)
NPX1=NPX(1)
IF((NC-IDX*10).EQ.1)IDX=-1
IF((NC-IDX*10).EQ.2)IDX=-2
CALL LAGRAN(XA,TABX(NPX1),TABY(NPX1),IDX+1,Y)
RETURN
END
```

8.2 DISER1 Subroutine

The subroutine DISER1 returns the array location of the lower bound value of the interval in which the independent variable lies. DISER1 is a modification of a MCAUTO library subroutine named DISSER.

The arguments for the DISER1 subroutine are as follows:

Subroutine DISER1 (XA, TAB, I, NX, ID, NPX, IND)

XA	- Independent variable
TAB	- X array
I	- Tabulated data location
NX	- Number of points in the independent array
ID	- Degree of interpolation
NPX	- Location of lower bound for data point XA, in the TAB array
IND	- Indicator

8.2.1 Solution Method. Not applicable

8.2.2 Assumptions. Not applicable

8.2.3 Computations. On entry of the independent variable, XA, and the tabulated data form the TABX array, DISER1 finds the tabulated data values that bound XA, and returns the smaller one to the calling program. If XA were to lie outside the lower end of the data, DISER1 would return the first data point as the lower bound. Should XA lie outside the upper tabulated value, the second from the last data point location is returned by DISER1.

8.2.4 Approximations. Not applicable

8.2.5 Limitations. Not applicable

8.2.6 DISER1 Variable Names

Variable	Description	Dimensions
IND	Solution indicator	-
I, ID, IT, J, NLOC, NLOW, NPB, NPT, NPU, NPX, NUPP, NX, NXX	Integer counters	-
Tab	Array of independent variables	-
XA	Independent variable	-

8.2.7 DISER1 Subroutine Listing

```
      SUBROUTINE DISER1(XA,TAB,I,NX,ID,NPX,IND)
      DIMENSION TAB(1)
      IF(XA-TAB(I))1,2,3
    1 IND=IND+1
      NPX=I
      RETURN
    2 XA=TAB(I)
      NPX=I
      RETURN
    3 J=I+NX-1
      IF(XA-TAB(J))6,5,4
    4 IND=IND+1
      NPX=J-ID
      RETURN
    5 XA=TAB(J)
      NPX=J-ID
      RETURN
    6 NPT=ID+1
      NPB=NPT/2
      NPU=NPT-NPB
      IF(NX-NPT)7,8,9
    7 ID=NX-1
      GO TO 6
    8 NPX=I
      RETURN
    9 NLOW=I+NPB
      NUPP=I+NX-(NPU+1)
      IF(NX-20)15,15,10
   10 NXX=NX/2+I
      IF(XA-TAB(NXX))11,14,12
   11 NXX=NXX-NX/4
      IF(XA-TAB(NXX))15,14,14
   12 NXX=NXX+NX/4
      IF(XA-TAB(NXX))13,14,14
   13 NLOW=NXX-NX/4
      GO TO 15
   14 NLOW=NXX
   15 DO 16 II=NLOW,NUPP
      NLOC=II
      IF(TAB(II)-XA)16,17,17
   16 CONTINUE
      NPX=NUPP-NPB+1
      RETURN
   17 NPX=NLOC-NPB
      RETURN
      END
```

8.3 LAGRAN Subroutine

The LAGRAN subroutine interpolates or extrapolates a data point from two known tabulated values. In addition, LAGRAN calculates viscosity using an ASTM viscosity equation and a fluid viscosity-pressure correction factor. The LAGRAN subroutine arguments are:

Subroutine LAGRAN (XA, X, Y, N, ANS)

XA - Independent variable

X - X array

Y - Y array

N - Solution Indicator

-1 = FLUIDF Calculation

0 = Viscosity Calculation

$\geq$1 = Degree of Interpolation

ANS - Dependent variable

8.3.1 Math Model. LAGRANGES interpolation equation is used in this subroutine to calculate the dependent variable. The LAGRANGE formula is:

$$P(x) = \sum_{i=0}^{m} L_i(x)\, y_i \qquad (1)$$

Where:

$L_i(x)$ is the Lagrange multiplier function.

$$L_i(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} \qquad (2)$$

The LAGRANGE equation generates a polynominal between two data points. The degree of the polynominal is that specified by the index value N. The dependent variable is returned as ANS in the subroutine argument.

An ASTM viscosity equation (See Appendix B) is used in the calculation of viscosity. The ASTM charts are based on this equation.

$$\mathrm{LOG}\,[\mathrm{LOG}\,(\nu+c)] = A - B\,\mathrm{LOG}\,T \qquad (3)$$

Where:

c = a constant

T = Temperature, °RANKINE

ν = Viscosity, CENTISTOKES

A,B = Constants for each fluid

LOG = Log to the base 10

The computational form of equation (3) is explained in Appendix B along with the mathematical formulation for the FLUIDF term.

8.3.2 Assumptions. The Lagrangian equation generated by the subroutine only uses the data points around the dependent variable to generate a polynominal for interpolation. The last or first set of two data points is used for extrapolation.

8.3.3 Computation. The procedure LAGRAN performs, whether it be interpolation the viscosity or the FLUIDF calculation, is always recognized by testing the N argument in the subroutine statement. If N is equal to minus one, then the FLUIDF factor is calculated. For N equal to zero the viscosity is computed. Otherwise N specifies the degree of interpolation to be used by the Lagrange formula. All results are returned to the calling program through the variable named ANS. The LAGRAN interpolation is a direct application of equation (1) to the given data.

Before evaluting the viscosity equation (3) for the viscosity value at XA temperature, the constants A and B must be calculated. They are solved using the data points that surround the dependent variable, or the first or last set of two data points if the dependent variable lies outside the range of the tabulated data. With the constants calculated for this fluid the viscosity can be computed from Equation (3).

8.3.4 Approximations - See Appendix B for a more thorough discussion on the approximations made for the viscosity and FLUIDF computations.

8.3.5 Limitations - Since the Lagrange method only uses two data points to interpolate it can become inaccurate for remotely spaced tabulated data points. Any degree of interpolation greater than two can lead to erroneous results.

8.3.6 LAGRAN Variable Names

Variable	Description	Dimensions
A	Constant for viscosity	-
ANS	Dependent variable	-
B	Constant for viscosity	-
I,J	Integer counters	-
N	Method of solution	-
	-1 = FLUIDF Calculation	
	0 = Viscosity calculation	
	>0 = Degree of interpolation	
PROD	Lagrange partial product	-
P1	LOG LOG of (Y(1) + C)	CENTISTOKES
P2	LOG LOG of (Y(2) + C)	CENTISTOKES
T1	LOG of T(1)	°R
T2	LOG of T(2)	°R
X	X-array	-
XA	Independent variable	-
Y	Y-array	-

The other data variables are explained in Appendix B.

8.3.7 LAGRAN Subroutine Listing

```
      SUBROUTINE LAGRAN(XA,X,Y,N,ANS)
      DIMENSION X(1),Y(1)
      IF(N.EQ.-1)GO TO 6
      IF(N.EQ.0)GO TO 4
      SUM=0.0
      DO 3 I=1,N
      PROD=Y(I)
      DO 2 J=1,N
      A=X(I)-X(J)
      IF(A) 1,2,1
    1 B=(XA-X(J))/A
      PROD=PROD*B
    2 CONTINUE
    3 SUM=SUM+PROD
      ANS=SUM
      RETURN
C     VISCOSITY CALCULATION
    4 CONTINUE
      A1=0.
      IF(Y(1).LE.2.)A1=EXP(-1.47-1.84*Y(1)-.51*Y(1)**2)
      A2=0.
      IF(Y(2).LE.2.)A2=EXP(-1.47-1.84*Y(2)-.51*Y(2)**2)
      P1=ALOG10(ALOG10(Y(1)+.7+A1))
      P2=ALOG10(ALOG10(Y(2)+.7+A2))
      T1=ALOG10(X(1)+460.)
      T2=ALOG10(X(2)+460.)
      B=(P1-P2)/(T2-T1)
      A=P1+B*T1
      Z=10**(10**(A-B*ALOG10(XA+460.)))
      IF(Z.LE.2.7)GO TO 5
      ANS=Z-.7
      RETURN
    5 ANS=(Z-.7)-EXP(-.7487-3.295*(Z-.7)+.6119*(Z-.7)**2
     A-.3193*(Z-.7)**3)
      RETURN
C     FLUIDF CALCULATION
    6 CONTINUE
      P1=ALOG10(ALOG10(Y(1)+.6))
      P2=ALOG10(ALOG10(Y(2)+.6))
      T1=ALOG10(X(1)+460.)
      T2=ALOG10(X(2)+460.)
      B=(P1-P2)/(T2-T1)
      A=P1+B*T1
      V0=10**(10**(A-B*ALOG10(XA+460.)))-.6
```

8.3.7 (Continued)

```
      T5=10**((.125989+A)/B)
      T1000=10**((-.477159+A)/B)
      S=ALOG10((T5)/100.+1.)-ALOG10((T1000)/100.+1.)
      DELX=65.10979*S
      S=6.65/DELX
      IF(S.LT..6)S=.6
      IF(S.GT.1.07)S=1.07
      ALPHA=3.23523-11.3886*S+13.1735*S*S-4.8881*S*S*S
      BETA=-5.33425+19.9521*S-23.9448*S*S+10.155*S*S*S
      CHI=3.35452-13.1273*S+17.1712*S*S-7.6551*S*S*S
      ANS=ALPHA+BETA*ALOG10(V0)+CHI*(ALOG10(V0))**2
      IF(ANS.LT.0.)ANS=0
      RETURN
      END
```

8.4 SIMULT SUBROUTINE

SIMULT is a subroutine for in-core solution of large, sparse systems of linear equations (Reference A8). The sub-program employs minimum row minimum column elimination. A limited number of zeros is stored and trivial arithmetic is used to preserve computer storage and to reduce the time required for solution. SIMULT is used in conjunction with CALC to obtain the system flows and pressures.

8.4.1 Solution Method

Excellent discussions on the Gauss-Jordan elimination technique can be found in many numerical analysis textbooks. Briefly the method is based on the three elementary row operations:

1. Interchange of any two rows.
2. Multiplication of a row by a scalar.
3. Addition of a multiple of one row to another row.

For example by applying a sequence of row transformations to a system of simultaneous equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m &= b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mm}x_m &= b_m \end{aligned}$$

Expressed in augmented form

$$\left[\begin{array}{cccc:c} a_{11} & a_{12} & \cdot & a_{1m} & b_1 \\ a_{21} & a_{22} & \cdot & a_{2m} & b_2 \\ a_{m1} & a_{m2} & \cdot & a_{mm} & b_m \end{array}\right]$$

Yields

$$\left[\begin{array}{c:c} I & X \end{array}\right]$$

where I is the identity matrix and X is the solution. The Gauss-Jordan elimination technique used in SIMULT requires elimination of only the elements in the upper or lower triangular partition of the array which is followed by a back substitution to obtain the solution.

Three arrays are generated that contain the number of non-zero elements in each row (IRENT()), the number of non-zero elements in each column (ICENT()) and the column number of each non-zero element (ICOL()) of an N x N array. These arrays are updated each time an element is eliminated or generated, so that the current row and column count and element location are available for pivot selection and row addition.

In SIMULT the IRENT array is searched to find the row with the least number of non-zero coefficients that has not been previously selected as the pivotal row. Should two or more rows satisfy this criteria, the row with the smallest row index is selected. Next the ICENT array is searched to select the column with least number of entries. In the event that two or more columns contain the same number of elements, the column with the smallest index is selected as the pivotal column.

Each row-column selection is thus used in the back substitution to obtain the solution.

8.4.2 Assumptions - Not applicable.

8.4.3 Computation - Upon entry into SIMULT the number of non-zero elements in each column and row of the M x M solution matrix is stored in ICENT( ) and IRENT( ). At this point the remainder of the program is contained within three nested loops. The outer loop selects a new pivotal element on each pass. The pivot element is stored in the order array for future use during subsequent iterations iterations in the CALC program. This is a time saving device to eliminate the necessity of selecting the same sequence of pivot elements on each iteration. Once the pivotal element has been selected, the pivotal row is normalized by dividing the row by the pivotal element. Since the pivotal element is normalized, it is set to one as a precaution against round-off errors.

The second loop is entered, which involves a row-by-row search for rows containing elements in the pivotal column. If the number of entries in the pivotal column has been reduced to one entry, there is no need to continue and the program selects a new pivotal element. Also if the pivotal row is selected all further tests are bypassed and the next row is selected since operations on the pivotal row are not permitted.

Finally, the inner loop is a column-by-column search of each row to determine if the row contains the pivotal element. At this point, there are three alternatives available:

1. If the column index in the row being searched is less than the pivotal column, it is necessary to continue searching the row.

2. If the column index is greater than the pivotal column, the row does not contain the pivotal column and a new row must be selected.

3. If the column index is equal to the pivotal column, the row contains the pivotal element and the row can be operated on by the pivotal row.

If the conditions in 3 are met, the pivotal row is multiplied by the negative of the element in the pivotal column of the row being operated on. Then the two rows are added. The element being eliminated is simply dropped from consideration by moving all entries to its right one space to the left. All elements remaining in the row are compared to ZTEST to see if any elements other than the element in the pivotal column were eliminated. If so, the row was further compressed to eliminate the zero entry from the row. Finally, the row is tested to see if the row count is zero which indicates a singularity. If a singularity is encountered an error message is printed:

* SINGULAR MATRIX-NO SOLUTION*

If a singularity is not encountered, the program continues looping until a pivotal element has been selected from each row and column, at this point, the solution vector is stored in the CALC2 array in a scrambled order. The solution is then unscrambled and stored in numerical order in the first column of the A( ) array.

8.4.4 Approximations - In situations where it is known that an operation will result in a zero or a one, the arithmetic operation is bypassed and the element simply set to zero or one.

8.4.5 Limitations - SIMULT is set up to solve only sparse systems of linear equations.

8.4.6 SIMULT Variable Names

Variable	Description	Dimension
ATEST, C	Dummy variables	--
CALC1( )	Matrix of coefficients	--
CALC2( )	NU matrix of constants	--
IC,II,IK	Dummy variables	--
ICENT( )	Array containing number of non-zero elements in each column of CALC1	--
ICOL( )	Array containing column location of each non-zero element of CALC1	--
IORDER( )	Array giving pivot selection based on min-row min-column criteria	--
IRENT( )	Array containing number of non-zero elements in each row of CALC1	--
ITER	Iteration count	--
IX,IY,J,JKL,JKOP, JKPI	Dummy variables	--
JCENT	Array identifying the number of non-zero entries in each column of CALC1	--

Variable	Description	Dimension
JCOL	Array identifying the non-zero filled columns of CALC1; the rows correspond to the rows of CALC1; elements in each row correspond to column number in each row of CALC1	--
JRENT	Array identifying the number of non-zero entries in each row of CALC1	--
LKJ,NAA,NK	Dummy variables	--
NPQ	Total number of rows in array	--
NU	Number of equations	--
OPROW	Dummy variable	--
PIVCOL	Pivot column	--
PIVROW	Pivot row	--
X,ZTEST	Dummy variables	--

8.4.7 SIMULT Subroutine Listing

```
       SUBROUTINE SIMULT(NU,ITER,CALC1,CALC2,JCOL,JRENT,JCENT,ICENT,
      AIRENT,IORDER,ICOL,NPQ)
       DOUBLE PRECISION C,X
       INTEGER PIVROW,PIVCOL,OPROW
       DIMENSION CALC1(1),CALC2(1),JCOL(1),JRENT(1),JCENT(1),ICOL(1)
       DIMENSION ICENT(1),IRENT(1),IORDER(1)
       ZTEST=0.0
       NAA=9
C        BUILD IRENT, ICENT, AND ICOL
       DO 1 I=1,NU
       IRENT(I)=JRENT(I)
       ICENT(I)=JCENT(I)
       DO 1 J=1,5
     1 ICOL(I+(J-1)*NPQ)=JCOL(I+(J-1)*NPQ)
       DO 24 LKJ=1,NU
       IF(ITER.NE.1) GO TO 4
       IK=100000
       DO 2 I=1,NU
       IF(IRENT(I).GE.IK.OR.IRENT(I).LE.0)GO TO 2
       PIVROW=I
       IK=IRENT(I)
     2 CONTINUE
       IORDER(LKJ)=PIVROW
       IK=100000
       IC=IRENT(PIVROW)
       DO 3 I=1,IC
       II=ICOL(PIVROW+(I-1)*NPQ)
       IF(ICENT(II).GE.IK.OR.ICENT(II).LE.0)GO TO 3
       PIVCOL=II
       IK=ICENT(II)
       IY=I
     3 CONTINUE
       IORDER(LKJ+NPQ)=PIVCOL
       IORDER(LKJ+2*NPQ)=IY
       GO TO 5
     4 PIVROW=IORDER(LKJ)
       PIVCOL=IORDER(LKJ+NPQ)
       IY=IORDER(LKJ+2*NPQ)
     5 X=CALC1(PIVROW+(IY-1)*NPQ)
       IC=IRENT(PIVROW)
       DO 6 J=1,IC
     6 CALC1(PIVROW+(J-1)*NPQ)=CALC1(PIVROW+(J-1)*NPQ)/X
       CALC1(PIVROW+(IY-1)*NPQ)=1.0
       CALC2(PIVROW)=CALC2(PIVROW)/X
       DO 22 I=1,NU
       IF(ICENT(PIVCOL).EQ.1)GO TO 23
       IF(I.EQ.PIVROW) GO TO 22
       IC=IABS(IRENT(I))
       DO 21 J=1,IC
       IF(ICOL(I+(J-1)*NPQ)-PIVCOL) 21,7,22
     7 OPROW=I
       JKOP=1
       JKPI=1
       C=-CALC1(OPROW+(J-1)*NPQ)
```

8.4.7 (Continued)

```
      X=CALC2(PIVROW)*C+CALC2(OPROW)
       CALC2(OPROW)=X
    8 CONTINUE
      IF(ICOL(PIVROW+(JKPI-1)*NPQ).EQ.0) GO TO 22
      IF(ICOL(OPROW+(JKOP-1)*NPQ).EQ.0) GO TO 9
      IF(ICOL(PIVROW+(JKPI-1)*NPQ)-ICOL(OPROW+(JKOP-1)*NPQ)) 9,12,
     120
    9 IRENT(I)=IRENT(I)+1
      IF(IRENT(I).LE.0) IRENT(I)=IRENT(I)-2
      II=IABS(IRENT(I))
      IF(II.GT.NAA)WRITE(6,10)II
   10 FORMAT(10X,*EXCEEDED MAX COLUMN NUMBER*,I10)
      IF(II.GT.NAA)STOP
      JKL=JKOP+1
   11 CONTINUE
      IX=II-1
      CALC1(OPROW+(II-1)*NPQ)=CALC1(OPROW+(IX-1)*NPQ)
      ICOL(OPROW+(II-1)*NPQ)=ICOL(OPROW+(IX-1)*NPQ)
      II=IX
      IF(II.GE.JKL) GO TO 11
      X=CALC1(PIVROW+(JKPI-1)*NPQ)*C
      CALC1(OPROW+(JKOP-1)*NPQ)=X
      ICOL(OPROW+(JKOP-1)*NPQ)=ICOL(PIVROW+(JKPI-1)*NPQ)
      IX=ICOL(OPROW+(JKOP-1)*NPQ)
      ICENT(IX)=ICENT(IX)+1
      GO TO 19
   12 IX=ICOL(OPROW+(JKOP-1)*NPQ)
      IF(IX.EQ.PIVCOL) GO TO 13
      X=CALC1(PIVROW+(JKPI-1)*NPQ)*C+CALC1(OPROW+(JKOP-1)*NPQ)
      CALC1(OPROW+(JKOP-1)*NPQ)=X
      ATEST=DABS(X)-ZTEST
      IF(ATEST.GT.0.0)GO TO 19
   13 ICENT(IX)=ICENT(IX)-1
      IRENT(OPROW)=IRENT(OPROW)-1
      IF(IRENT(OPROW))16,14,17
   14 CONTINUE
      WRITE(6,15)
   15 FORMAT(10X,*SINGULAR MATRIX-NO SOLUTION*)
      STOP
   16 CONTINUE
      IRENT(OPROW)=IRENT(OPROW)+2
   17 IX=IABS(IRENT(OPROW))
      DO 18 NK=JKOP,IX
      CALC1(I+(NK-1)*NPQ)=CALC1(I+NK*NPQ)
   18 ICOL(I+(NK-1)*NPQ)=ICOL(I+NK*NPQ)
      IX=IX+1
      ICOL(I+(IX-1)*NPQ)=0
      JKPI=JKPI+1
      GO TO 8
   19 JKPI=JKPI+1
   20 JKOP=JKOP+1
      GO TO 8
   21 CONTINUE
   22 CONTINUE
   23 CONTINUE
```

8.4.7 (Continued)

```
      IRENT(PIVROW)=-IRENT(PIVROW)
      ICENT(PIVCOL)=-ICENT(PIVCOL)
   24 CONTINUE
      DO 25 I=1,NU
      I1=ICOL(I)
   25 CALC1(I1)=CALC2(I)
      RETURN
      END
```

8.5 VISD Subroutine

The VISD subroutine controls the computation of the fluid viscosity, density, and a viscosity pressure correction factor at the system operating temperature. The fluid density and viscosity are also calculated for a temperature of 100°F. The VISD subroutine directs the proper data to the INTERP subroutine which actually handles the calculation procedure.

8.5.1 Math Model - Not applicable

8.5.2 Assumptions - Not applicable

8.5.3 Computations - The first call to INTERP returns with the fluid density (DENS) at the system operating temperature. The next call gives the density at 100°F (D100). The third call to the INTERP subroutine yields the FLUIDF term which is a viscosity-pressure correction factor. A DO loop is set up with the number of viscosity data points NVIS at the upper parameter. If the system temperature is the same as the input viscosity data temperature input by the user, the viscosity is taken directly from the input data. The same applies if the 100°F temperature is an input point. If either temperature cannot be found in the input data, INTERP is called to compute the appropriate viscosities. Should a viscosity value ever be a negative number, which could result from erroneous data input, IERROR is set to one and program control is passed to SFAN where an error message is output.

8.5.4 Approximations - Not applicable

8.5.5 Limitations - The parameters computed for the FLUIDF term rely on the user input viscosity-temperature data. An erroneous FLUIDF factor may be calculated if extrapolation outside the data range is required.

For any viscosity-pressure correction factors that are computed as negative values, FLUIDF is set to zero and thus there are no pressure correction terms used in the program.

8.5.6 VISD Variable Names

Variable	Description	Dimensions
I	Current Viscosity Data Point	-
IERROR	Error Indicator 0 = No error 1 = Program Termination	-
IND	Interpolation Indicator 0 = Normal Interpolation 1 = Extrapolation Outside the range of data points	-

The data variables stored in labeled common are explained in the Block Data section of the Main Program description.

8.5.7 VISD Subroutine Listing

```
      SUBROUTINE VISD
C     REVISED  MARCH 3,1975
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      COMMON /BLK2/VVISC(9),VTEMP(9),DDENS(2),DTEMP(2),NVIS
      COMMON /BLK7/IERROR
      CALL INTERP(TEMP,DTEMP,DDENS,10,2,DENS,IND)
      CALL INTERP(100.,DTEMP,DDENS,10,2,D100,IND)
      CALL INTERP(TEMP,VTEMP,VVISC,12,NVIS,FLUIDF,IND)
      DO 1 I=1,NVIS
      IF(TEMP.EQ.VTEMP(I))VISC=VVISC(I)
      IF(VTEMP(I).EQ.100.)V100=VVISC(I)
    1 CONTINUE
      IF(VISC.NE.0.)GO TO 2
      CALL INTERP(TEMP,VTEMP,VVISC,11,NVIS,VISC,IND)
    2 IF(V100.NE.0.)GO TO 3
      CALL INTERP(100.,VTEMP,VVISC,11,NVIS,V100,IND)
    3 IF(VISC.GT.0..AND.V100.GT.0.)RETURN
      IERROR=1
      RETURN
      END
```

8.6 DARR Subroutine

The DARR subroutine is used in conjunction with element type 10 data stored in BRANCHP array to provide the data points in two arrays for use by the INTERP subroutine. The data stored in BRANCHP array have to be relocated and put into a dummy array for use by the INTERP subroutine when a pressure drop is required from one of the special components.

8.6.1 Math Model

Not applicable.

8.6.2 Assumptions

Not applicable.

8.6.3 Computations

Two arrays DARR1 and DARR2 are dimensioned to the maximum number of data points for any one element in the special element array. That number is brought through the subroutine argument with the variable name M, when M is greater than 1. The row location in the BRANCHP array of the element is K. A DO loop is set up to take the first M data points from BRANCHP and insert them into the DARR1 array. Another DO loop places the next M data points into the DARR2 array. These two arrays with the special element data points are then passed back to the calling subroutine.

8.6.4 Approximations

Not applicable.

8.6.5 Limitations

Not applicable.

8.6.6 DARR Variable Names.

Variable	Description	Dimensions
BRANCHP	Dynamic element data storage array	--
DARR1	Dummy array containing M number of flow data points	GPM
DARR2	Dummy array containing M number of pressure data points	PSI
I,IG,IH,II,J,J1	Integer counters	--
K	Row location in PEC10 array	--
M	Number of data points	--
NBP2	Total length of BRANCHP array	--

The description of the information stored in each element array may be found in the individual element subroutines.

8.6.7 DARR Subroutine Listing

```
      SUBROUTINE DARR(DARR1,DARR2,M,K,BRANCHP,NBP2)
      DIMENSION BRANCHP(1),DARR1(M),DARR2(M)
      IG=M+6
      I=1
      DO 1 J=7,IG
      DARR1(I)=BRANCHP(K+J*NBP2)
    1 I=I+1
      I1=1
      IH=M+IG
      IG=IG+1
      DO 2 J1=IG,IH
      DARR2(I1)=BRANCHP(K+J1*NBP2)
    2 I1=I1+1
      RETURN
      END
```

8.7 VCHEK Subroutine

When the system is initially assembled, a flow direction is assumed for the check valve, one way restrictor and relief valve, element types 3, 31 and 33 respectively. The check valve flow is initially assumed in the free flow direction, the one way restrictor in the restricted flow direction and the relief valve in the relief flow direction, but with the valve closed. This is done for stability during the system balancing. When the system is balanced, a call to VCHEK checks to see that the assumed flow direction was correct. If the assumed flow direction was incorrect, the indicator is changed in the element data array to use resistance factors for flow in the other direction. The indicator, IC, is then given a value of 1 so that a rebalance of the system will again be done.

8.7.1 VCHEK Variable Names

Variables	Description	Dimensions
BLEG	General purpose array	--
BRANCHP	Array containing dynamic element data	--
I	Integer counter	--
IC	Indicator to rebalance system	--
ILEP	Array containing pressure point numbers at end of each leg	--
KC,LC	Integer counters	--
LEG	Leg number (row in BLEG)	--
MC	Integer counter	--
ML	Total number of legs in system	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG and ILEP arrays	--
NPQ	Total number of rows in PQL array	--
PQL	Array containing pressure point data	--
Q	Leg flow rate	--
Ty	Element type	--

8.7.2 VCHEK Subroutine Listing

```
  SUBROUTINE VCHEK(ML,IC,BRANCHP,NBP2,BLEG,NL,ILEP,PQL,NPQ)
  DIMENSION PQL(1),BRANCHP(1),BLEG(1),ILEP(1)
  MC=0
  LC=0
  KC=0
1 DO 6 I=1,NBP2
  TY=BRANCHP(I)
  IF(TY.EQ.0.)GO TO 7
  LEG=BRANCHP(I+20*NBP2)
  IF(TY.EQ.3.)GO TO 2
  IF(TY.EQ.31.)GO TO 3
  IF(TY.EQ.33.)GO TO 4
  GO TO 6
2 Q=BLEG(LEG+2*NL)
  IF(BRANCHP(I+7*NBP2).EQ.-1.)GO TO 6
  IF(Q.GT.0..AND.BRANCHP(I+6*NBP2).EQ.1.)GO TO 6
  IF(Q.LT.0..AND.BRANCHP(I+6*NBP2).EQ.2.)GO TO 6
  BRANCHP(I+7*NBP2)=-1.
  KC=KC+1
  GO TO 6
3 Q=BLEG(LEG+2*NL)
  IF(BRANCHP(I+12*NBP2).EQ.-1.)GO TO 6
  IF(Q.GT.0..AND.BRANCHP(I+10*NBP2).EQ.1.)GO TO 6
  IF(Q.LT.0..AND.BRANCHP(I+10*NBP2).EQ.2.)GO TO 6
  BRANCHP(I+12*NBP2)=-1.
  LC=LC+1
  GO TO 6
4 IF(BRANCHP(I+7*NBP2).EQ.-1.)GO TO 6
  IF(BRANCHP(I+7*NBP2).EQ.-1.)GO TO 6
  IF(Q.GT.0..AND.BRANCHP(I+6*NBP2).EQ.1.)GO TO 5
  IF(Q.LT.0..AND.BRANCHP(I+6*NBP2).EQ.2.)GO TO 5
  GO TO 6
5 IF(ABS(PQL(ILEP(I))-PQL(ILEP(I+NL))).LT.BRANCHP(I+5*NBP2))GO TO
 16
  BRANCHP(I+7*NBP2)=-1.
  MC=MC+1
6 CONTINUE
7 CONTINUE
  IF(MC.GT.0.OR.LC.GT.0.OR.KC.GT.0)IC=1
  RETURN
  END
```

8.8 ACTCHEK Subroutine

ACTCHEK subroutine is used to determine whether or not there is cavitation in an actuator. The system is allowed to initially balance with a negative pressure in an actuator. When ACTCHEK is called, a check is made to see if a pressure was negative. If it is the pressure is set to 0 psi and an indicator (IC = 1) is passed to CALC subroutine to indicate a system rebalance is necessary. Additionally, NPQL2 array is reset to include the additional fixed pressure point.

8.8.1 ACTCHEK Variable Names

Variables	Description	Dimensions
BRANCHP	Array containing dynamic element data	--
IC	Indicator to rebalance system	--
INDEX	Pressure point number	--
J,J1	Integer counters	--
KOUNT	Row location in PCHECK array	--
N	Branch point number of actuator	--
NBP2	Total number of rows in BRANCHP array	--
NE	Actuator extend pressure point number	--
NPQ	Total number of rows in PQL array	--
NPQL2	Array containing fixed pressure point numbers	--
NR	Actuator retract pressure point number	--
PCHECK	Dummy array for actuator data	--
PQL	Array containing pressure point data	--
TEST	Calculated internal actuator pressure	PSIG

8.8.2 ACTCHEK Subroutine Listing

```
      SUBROUTINE ACTCHEK(N,IC,BRANCHP,NBP2,PQL,NPQ)
C        CHECK FOR CAVITATION IN ACTUATORS
      DIMENSION BRANCHP(1),PQL(1)
      COMMON /BLK3/NPQL2(20),PQL2(20)
      DIMENSION PCHECK(25,5)
      KOUNT=0
      DO 3 J=1,NBP2
      IF(BRANCHP(J).NE.4.)GO TO 3
      IF (BRANCHP(J+13*NBP2).EQ.0.0)GO TO 4
C        NE,NR ARE PRESSURE POINTS FOR ACTUATOR
      NE=BRANCHP(J+3*NBP2)
      NR=BRANCHP(J+4*NBP2)
      IF (PQL(NE).GE.0.0)GO TO 1
C        CAVITATION ON EXTENDING SIDE
      KOUNT=KOUNT+1
      PCHECK(KOUNT,1)=NE
      PCHECK(KOUNT,2)=4.
      PCHECK(KOUNT 3)=PQL(NE)
      PCHECK(KOUNT,4)=1.
      PCHECK(KOUNT,5)=J
      WRITE(6,2)(PCHECK(KOUNT,K),K=1,5)
    1 IF (PQL(NR).GE.0.0)GO TO 3
C        CAVITATION ON RETRACTING SIDE
      KOUNT=KOUNT+1
      PCHECK(KOUNT,1)=NR
      PCHECK(KOUNT,2)=4.
      PCHECK(KOUNT,3)=PQL(NR)
      PCHECK(KOUNT,4)=-1.
      PCHECK(KOUNT,5)=J
      WRITE(6,2)(PCHECK(KOUNT,K),K=1,5)
    2 FORMAT(' ',15H PCHECK ARRAY  ,5F12.2)
    3 CONTINUE
    4 IF (KOUNT.EQ.0)RETURN
      TEST=0.
      DO 5 J=1,KOUNT
      IF (PCHECK(J,3).GT.TEST)GO TO 5
      TEST=PCHECK(J,3)
      INDEX=PCHECK(J,1)
    5 CONTINUE
      J1=N+1
      DO 6 J=J1,20
    6 NPQL2(J1)=0
      NPQL2(N+1)=INDEX
      PQL(INDEX)=0.001
      IC=1
      WRITE(6,7)(NPQL2(K),K=1,J1)
    7 FORMAT(' ',15H  NPQL2 ARRAY  ,5F12.2)
      RETURN
      END
```

8.9 FLOCHEK SUBROUTINE

Subroutine FLOCHEK is used to test the type 37 flow regulator and type 38 orifice sizer. The flow regulator is tested to see that the calculated pressure difference is greater than zero or the minimum input pressure. If the pressure difference is less, an indicator, IC=1, is passed back to CALC subroutine and another indicator is passed to BRANCHP subroutine. The BRANCHP indicator is used by the DVS034 dynamic calculation subroutine to indicate that a different method of calculation is required whether or not a system rebalance is necessary.

The orifice sizer section of the program calculates an orifice diameter for the input flow rate in the leg. The pressure drop for the orifice is calculated under dynamic conditions. If the input flow rate is too high and gives a negative pressure drop calculation, an error message is printed.

8.9.1 FLOCHEK Variable Names

Variables	Description	Dimensions
BLEG	General purpose array	--
BRANCHP	Array containing dynamic element data	--
DP	Pressure difference from inlet to outlet of flow regulator	PSID
DP1	Minimum pressure difference for rated flow through flow regulator	PSID
I	Integer counter	--
IC	Indicator to rebalance system	--
ILEP	Array containing pressure point numbers at end of each leg	--
IP1	Inlet pressure point number to flow regulator	--
IP2	Outlet pressure point number to flow regulator	--
ML	Total number of legs in system	--
NBP2	Total number of rows in BRANCHP array	--
NL	Total number of rows in BLEG and ILEP arrays	--
NPQ	Total number of rows in PQL array	--
PQL	Array containing pressure point data	--

8.9.2 FLOCHEK Subroutine Listing

```
      SUBROUTINE FLOCHEK(IC,BRANCHP,NBP2,BLEG,NL,PQL)
      DIMENSION PQL(1),BLEG(1),BRANCHP(1)
      COMMON /BLK1/TEMP,VISC,DENS,D100,PAMB,ALT,FLUIDF,FLUIDK,V100
      DO 6 I=1,NBP2
      TY=BRANCHP(I)
      IF(BRANCHP(I).EQ.0.)RETURN
      IF(TY.EQ.37..OR.TY.EQ.38.)GO TO 1
      GO TO 6
    1 IP1=BRANCHP(I+3*NBP2)
      IP2=BRANCHP(I+4*NBP2)
      DP=PQL(IP1)-PQL(IP2)
      DP1=BRANCHP(I+6*NBP2)
      LEG=BRANCHP(I+15*NL)
      IF(DP1.LE.DP)BLEG(LEG+2*NL)=BRANCHP(I+5*NBP2)
      IF(TY.EQ.38.)GO TO 2
      IF(DP1.LE.DP)GO TO 6
      IF(BRANCHP(I+8*NBP2).EQ.1.)GO TO 6
      BRANCHP(I+8*NBP2)=1.
      IC=1
      GO TO 6
    2 Q=BRANCHP(I+5*NBP2)
      IF(DP.LE.0.)GO TO 4
      PAVG=(PQL(IP1)+PQL(IP2))/2.
      DENP=(1+(PAVG/200000.))*DENS
      DIA=SQRT(Q/(236.*.6*SQRT(DP/DENP)))
      WRITE(6,3)BRANCHP(I+NBP2),DIA
    3 FORMAT(//' ORIFICE DIAMETER FOR JCT NO. ',F4.0,' = ',F5.3//)
      GO TO 6
    4 WRITE(6,5)Q,BRANCHP(I+NBP2)
    5 FORMAT(//' ','***ERROR***CANNOT SIZE ORIFICE FOR THE GIVEN',/,'FLO
     1RATE OF ',F8.2,' GPM AT JCT NO. ',F4.0,//)
    6 CONTINUE
      RETURN
      END
```

8.10 MTRCHK SUBROUTINE

Subroutine MTRCHK tests BRANCHP for hydraulic motors after a system balance is made. If the calculated flow through the motor is negative, this is an abnormal condition. An indicator is placed in BRANCHP to signal the dynamic calculation subroutine DMTR8 that a different calculation method is required. An indicator, IC=1, is passed to CALC subroutine to indicate a system rebalance is required.

8.10.1 MTRCHK Variable Names

Variable	Description	Dimension
BRANCHP	Dynamic element data storage array	--
I	Integer counter	--
IC	Indicator to CALC subroutine to rebalance system	--
NBP2	Total number of rows in BRANCHP array	--
Q	Flow rate	GPM
TY	Element type	--

8.10.2 MTRCHK Subroutine Listing

```
      SUBROUTINE MTRCHK(BRANCHP,NBP2,IC)
      DIMENSION BRANCHP(1)
      DO 1 I=1,NBP2
      IF(BRANCHP(I).EQ.0.)RETURN
      TY=BRANCHP(I)
      IF(TY.NE.8.)GO TO 1
      Q=BRANCHP(I+13*NBP2)
      IF(Q.LT.0.)BRANCHP(I+14*NBP2)=1.
      IF(Q.LT.0.)IC=1
    1 CONTINUE
      RETURN
      END
```

8.11 Subroutine GRAPH2 and SCALED

Subroutine GRAPH2 plots the selected junction and flow, pressure or actuator position versus time. The user can select the scales or the scales will be selected from SCALED subroutine.

8.11.1 GRAPH2 Variable Names

Variable	Description	Dimension
I	Integer counter	--
ICHART (-)	X and Y axis write characters	--
IPCHAR (1)	Plot character	--
ISP	Integer counter for Y axis	--
ISPACE (-)	Temporary variable for writing X and Y axis scales	--
JJ	Indicator for setting scales JJ = 0 Use subroutine SCALED JJ = 1 Use input valves	--
LINE	Integer counter for plot line number	--
NPLTPT	Number of points per plot	--
OMEGA (-)	Pump speed values	RPM
XAX	Temporary variable for writing X axis scale values	--
XDELTA	X-axis scale increment value	--
XMAX	Last (highest) X axis value	--
XMIN	First (lowest) X axis value	RPM
Y	Temporary variable - Y axis scale value	--
YDELTA	Y-axis scale incremental value	--
YLAST	Last Y-axis scale value	--
YLO	Lowest value in YPLT search range	--
YMAX	Maximum value to be plotted	--
YMIN	Minimum value to be plotted	--

Variable	Description	Dimension
YPLT (-,-)	Array of value to be plotted	--
YUP	Highest value in YPLT search range	--

SCALED Variable Names

Variable	Description	Dimension
AMAX	Maximum value to be plotted	--
AMIN	Minimum value to be plotted	--
IBOT	Variable used to calculate Y axis scale values	--
IEMAX	Variable used to calculate Y axis scale values	--
IEXP	Variable used to calculate Y axis scale values	--
ITOP	Variable used to calculate Y axis scale values	--
J	Integer counter	--
MANT	Variable used to calculate Y axis scale values	--
RANGE	Range of values to be plotted	--
RMAX	Maximum Y axis scale value	--
RMIN	Minimum Y axis scale value	--
SCALE (-)	Scale factors for Y axis	--

8.11 2 GRAPH2 Subroutine Listing

```
      SUBROUTINE GRAPH2(OMEGA,YPLT,NPLTPT,IPCHAR,JJ,YMIN,YMAX,XMIN,XMAX)
C
C
      DIMENSION OMEGA(125),YPLT(125),ISPACE(101),
     1XAX(6),ICHART(4),IPCHAR(1)
      DATA ICHART/1HI,1H-,1H+,1H /
C-----SCALE X
      IF(JJ.EQ.1)GO TO 3
      XMAX=OMEGA(1)
      XMIN=XMAX
      DO 1 I=2,NPLTPT
      XMAX=AMAX1(XMAX,OMEGA(I))
    1 XMIN=AMIN1(XMIN,OMEGA(I))
      IF(XMAX.NE.XMIN)GO TO 2
      XMAX=XMAX+50.
      XMIN=XMIN-50.
    2 CALL SCALED(XMAX,XMIN)
    3 XDELTA=(XMAX-XMIN)/100.
      IF(JJ.EQ.1)GO TO 6
C------FIND Y-PARAMETERS
    4 YMAX=YPLT(1)
      YMIN=YMAX
      DO 5 I=2,NPLTPT
      YMAX=AMAX1(YMAX,YPLT(I))
    5 YMIN=AMIN1(YMIN,YPLT(I))
      CALL SCALED(YMAX,YMIN)
      IF(YMAX.NE.YMIN)GO TO 6
      YMAX=YMAX+25.
      YMIN=YMIN-25.
    6 YDELTA=(YMAX-YMIN)/50.
C-----ADVANCE TO TOP OF NEXT PAGE
      WRITE(6,7)
    7 FORMAT (1H1)
C-----LOOP FOR EACH PLOT LINE
      Y=YMAX + YDELTA
    8 DO 27 LINE=1, 51
      YLAST=Y
      Y=Y-YDELTA
      YUP=Y+YDELTA/2.
      YLO=YUP-YDELTA
C-----FIRST + LAST CHARACTER ON LINE = *I*
      ISPACE(1)=ICHART(1)
      ISPACE(101)=ICHART(1)
C-----FIRST + LAST LINES ALL *-*, EXCEPT *+* IN COL. 11,21,31,...81,91
        IF(LINE.NE.1.AND.LINE.NE.51) GO TO 12
    9 DO 11 ISP=2,100
      IF((ISP-1).EQ.(ISP-1)/10*10)GO TO 10
      ISPACE(ISP)=ICHART(2)
      GO TO 11
   10 ISPACE(ISP)=ICHART(3)
```

8.11.2 (Continued)

```
   11 CONTINUE
      GO TO 17
C-----INITIALIZE COL.2-100 OF LINES 2-50 TO * *, OR **-------*--* IF AX
   12 IF(Y.LE.O..AND.YLAST.GT.O.)GO TO 15
   13 DO 14 ISP=2,100
   14 ISPACE(ISP)=ICHART(4)
      GO TO 17
   15 DO 16 ISP=2,100
      ISPACE(ISP)=ICHART(2)
   16 IF((ISP-1).EQ.(ISP-1)/10*10)ISPACE(ISP)=ICHART(3)
C-----SEARCH YPLT FOR VALUES IN RANGE YLO.LT.YPLT.GE.YUP
   17 DO 23 I=1, NPLTPT
      IF(YPLT(I).GT.YLO .AND. YPLT(I).LE.YUP)GO TO 18
      GO TO 23
C-----FIND COLUMN NEAREST TO I-TH VALUE OF OMEGA
   18 ISP=(OMEGA(I)-XMIN)/XDELTA + 1.5
      IF(ISP) 23,19,20
   19 ISP=1
      GO TO 22
   20 IF(ISP-102)22,21,23
   21 ISP=101
   22 ISPACE(ISP)=IPCHAR(1)
   23 CONTINUE
C-----LINES 1,11,... 41,51 HAVE Y-VALUES---THESE, PLUS LINES 6,16,26,3
C-----+ 46 ALSO HAVE *+* IN COL 1+101 IF NO PLOT CHARACTER PRESENT
      IF((LINE-1).NE.(LINE-1)/5*5)GO TO 25
      IF(ISPACE(1).NE.IPCHAR(1))ISPACE(1)=ICHART(3)
      IF(ISPACE(101).NE.IPCHAR(1))ISPACE(101)=ICHART(3)
      IF((LINE-1).NE.(LINE-1)/10*10)GO TO 25
C-----WRITE PLOT LINE
      WRITE(6,24)Y, ISPACE
   24 FORMAT(' ',1X,17X,F9.2,2X,101A1)
      GO TO 27
   25 WRITE(6,26)ISPACE
   26 FORMAT(' ',1X,28X,101A1)
   27 CONTINUE
C-----CALCULATE + PRINT X-AXIS VALUES
   28 DO 29 I=1, 6
   29 XAX(I)=XMIN + (I-1)*20.*XDELTA
      WRITE(6,30) XAX
   30 FORMAT(' ',1X,22X,5(F9.2,11X),F9.2)
      RETURN
      END
```

8.11.2 (Continued)

SCALED Subroutine Listing

```
      SUBROUTINE SCALED(RMAX,RMIN)
      DIMENSION SCALE(6)
      DATA SCALE/.5,1.,2.,5.,10.,20./
      RANGE=RMAX-RMIN
      AMAX=RMAX
      AMIN=RMIN
      IEXP=ALOG10(RANGE)
      MANT=RANGE/10.**IEXP
      IF(RANGE.GT.MANT*10.**IEXP)MANT=MANT+1
      GO TO (2,3,4,4,4,5,5,5,5,1),MANT
    1 MANT=1
      IEXP=IEXP+1
    2 J=2
      GO TO 6
    3 J=3
      GO TO 6
    4 J=4
      GO TO 6
    5 J=5
    6 IEMAX=ALOG10(RMAX)
      RMAX=INT(AMAX/10.**IEMAX)*10.**IEMAX
    7 IF(RMAX.GE.AMAX)GO TO 8
      RMAX=RMAX+.05*SCALE(J)*10.**IEXP
      GO TO 7
    8 RMIN=RMAX-SCALE(J)*10.**IEXP
      IF(RMIN.LE.AMIN)GO TO 9
      J=J+1
      IF(J.LT.5.5)GO TO 8
      J=1
      IEXP=IEXP+1
      GO TO 8
    9 IF(RMIN*AMIN.GT.0.)GO TO 10
      RMIN=0.
      RMAX=SCALE(J)*10.**IEXP
      IF(RMAX.LT.SCALE(J-1)*10.**IEXP)RMAX=SCALE(J-1)*10.**IEXP
   10 IF(RMIN.LT.0.)GO TO 11
      IF(RMIN.GT..1*RMAX)GO TO 11
      RMIN=0.
      RMAX=SCALE(J)*10.**IEXP
      IF(RMAX.LT.AMAX)RMAX=SCALE(J+1)*10.**IEXP
      RETURN
   11 ITOP=(RMAX-AMAX)/(.05*SCALE(J)*10.**IEXP)
      IBOT=(AMIN-RMIN)/(.05*SCALE(J)*10.**IEXP)
      IF(ITOP.EQ.IBOT)RETURN
      RMIN=RMIN+IABS((ITOP-IBOT)/2)*.05*SCALE(J)*10.**IEXP
      RMAX=RMIN+SCALE(J)*10.**IEXP
      RETURN
      END
```

8.12 QTCALC Subroutine

Subroutine QTCAIC is used when running in the quasi transient mode.

8.12.1 Math Model - Entry to QTCALC is based on whether or not a time interval was placed in QT15 array. Upon entry, an initial calculation is made to determine the time step required to produce 101 points for a graph output plot. If no time step is input, a default value of 100 time steps (101 data points) is used. If less than 100 steps are called for, the graph plot will be only a partial plot. If more than 100 steps are required, based on the input time interval and time step data, only 101 points will be saved and plotted. These points are spaced throughout the time interval and the storage location for the graph plot is given as the integer truncation plus 1 as follows:

$$\text{NTSP} = \frac{\text{Number of Time Step} \times 100}{\text{Total Number of Ti}\quad\text{Steps}} + 1$$

For example if preliminary calculation showed that 280 time steps would be taken during the quasi-transient calculations and the 13th time step was taken, the graph storage location would be only the 5th.

$$\text{NTSP} = \frac{13 \times 100}{280} + 1 = 5$$

All calculated time steps are output in tabulated form even though only selected points may be output in graph form. The data for the quasi-transient run is initialized before the first calculation is made. A check is made of the quasi-transient data arrays (QT16, QT17 and QT18) to determine initial component positions and other parameters. As each time step is taken, the component element parameters are updated before the new calculation is made. The update of the parameters are made to the element data contained in BRANCHP array only. Data in arrays

QT16, QT17 and QT18 are used only to determine which data is to be placed in BRANCHP array positions.

8.12.2 Computations

The system is initially balanced at time zero (t = 0). Then, time is increased one time step. Using the initial load conditions and valve positions, the system volume changes at actuators and accumulators are calculated to give a pertubation to the system.

$$Q_o \times \Delta t = \Delta \text{ Volume}_{0-1}$$

New positions for actuators and accumulators are calculated. Using the new positions, average loads and pressures are calculated for the time step.

$$\text{LOAD}_{\text{AVG}} = \frac{\text{LOAD}_0 + \text{LOAD}_1}{2} \qquad P_{\text{AVG}} = \frac{P_0 + P_1}{2}$$

The system is balanced again using the average conditions of load and pressure. Final volume changes are calculated which give the system volume changes for the time step. If a valve position signal changes or an actuator piston bottoms somewhere during the time step, the calculation is used only for that part of the time step. Values are reinitialized and calculation is made for the remainder of the time step. This procedure is followed until all time steps have been taken. At the beginning of each time step flows and pressures are initialized; therefore a minimum of two calls to CALC subroutine are made. A description of quasi-transient data output is given in Section VII.

8.12.3 QTCALC Variable Names

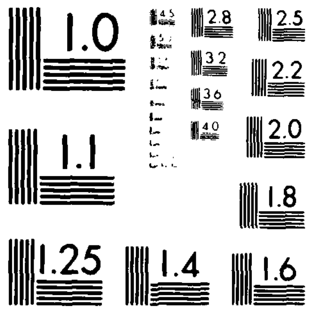
Variable	Description	Dimensions
BLEG	General purpose array	--
BRANCHP	Dynamic element data storage array	--
CALC1	Matrix of coefficients	--
CALC2	NU matrix of constants	--
ICENT	Array containing number of non-zero elements in each column of CALC1	--
ICOL	Array containing column location of each non-zero element of CALC1	--
IDIAG	Array which identifies which columns of CALC1 contain positive conductance values	--
ILEP	Array of leg numbers with up and downstream pressure points	--
INEG	Array which stores the second appearance of a negative conductance value	--
IORDER	Array giving pivot selection based on min-row min-column criteria	--
IPQL2	Integer counter	--
IRENT	Array containing number of non-zero elements in each row of CALC1	--
JCENT	Array identifying the number of non-zero entries in each column of CALC1	--
JCOL	Array identifying the non-zero filled columns of CALC1; the rows correspond to the rows of CALC1; elements in each row correspond to column number in each row of CALC1	--
JEM	Total number of pressure points in the system	--
JNEG	Array which identifies which column in CALC1 contains the first appearance of a negative conductance value in CALC1 array	--

8.12.3 (Continued)

Variable	Description	Dimensions
NBP	Total number of elements in BRANCHP array	--
NBP2	Total length of BRANCHP array	--
NL	Total length of BLEG & ILEP array	--
NPQ	Total number of rows in PQL array	--
NPQL2	Array containing row location in BRANCHP array of element with fixed pressure	--
N16	Length of QT16 array	--
N17	Length of QT17 array	--
N18	Length of QT18 array	--
N19	Length of QT19 array	--
PAMB	Atmospheric ambient pressure	PSI
PQL	Array containing calculated pressures, element types and junction numbers	--
PQL2	Array of constant pressure point junction	--
PRINT	Array containing output types	--
QT15	Storage array for quasi-transient temperatures	--
QT16	Storage array for additional element which used in quasi-transient calculations	--
QT17	Storage array for quasi-transient valve data	--
QT18	Storage array for quasi-transient valve data	--
QT19	Storage array to indicate quasi-transient output	--

8.12.4 QTCALC Subroutine Listing

```
  SUBROUTINE QTCALC(BRANCHP,NBP2,PQL,NPQ,BLEG,ILEP,NL,QT15,QT16,N16,
 1QT17,N17,QT18,N18,QT19,N19,ML,N,JEM,CALC1,JCOL CALC2,
 2JRENT,JCENT,IDIAG,JNEG,INEG,IRENT,ICENT,IORDER ICOL,PQL2,NPQL2)
  DIMENSION AR1(7),AR2(7)
  DIMENSION CALC1(1),CALC2(1),JCOL(1),JRENT(1),JCENT(1),ICOL(1)
  DIMENSION IDIAG(1),JNEG(1),INEG(1),BRANCHP(1)
  DIMENSION ICENT(1),IRENT(1),IORDER(1)
  DIMENSION BLEG(1),ILEP(1),PQL(1)
  DIMENSION PQL2(1),NPQL2(1)
  DIMENSION QT15(1),QT16(1),QT17(1),QT18(1),QT19(1)
  TI=QT15(1)
  TF=QT15(2)
  DT=QT15(3)
  IF(DT.EQ.0.)DT=(TF-TI)/100.
  TS=(TF-TI)/DT
  ITS=TS
  DO 9 I=1,1000
  IF(QT16(I).EQ.0.)GO TO 10
  IF(QT16(I).EQ.4.)GO TO 1
  IF(QT16(I).EQ.7.)GO TO 6
  GO TO 9
1 AJCT=QT16(I+N16)
  DO 2 J=1,NBP2
  IF(BRANCHP(J).EQ.0.)GO TO 9
  IF(BRANCHP(J).EQ.4..AND.BRANCHP(J+NBP2).EQ.AJCT)GO TO 3
2 CONTINUE
  GO TO 9
3 STRP=BRANCHP(J+10*NBP2)
  NP=QT16(I+2*N16)
  DO 4 K=1,7
  AR1(K)=0.
4 AR2(K)=0.
  DO 5 NAR=1,NP
  AR1(NAR)=QT16(I+(2+NAR)*N16)
5 AR2(NAR)=QT16(I+(2+NAR+NP)*N16)
  CALL INTERP(STRP,AR1,AR2,10,NP,ALOAD,1)
  BRANCHP(J+10*NBP2)=ALOAD
  GO TO 9
6 AJCT=QT16(I+N16)
  DO 7 J=1,NBP2
  IF(BRANCHP(J).EQ.0.)GO TO 9
  IF(BRANCHP(J).EQ.7.).AND.BRANCHP(J+NBP2).EQ.AJCT)GO TO 12
7 CONTINUE
  GO TO 9
8 AVOL=QT16(I+2*N16)
  APRESS=QT16(I+3*N16)
  AMOV1=BRANCHP(J+9*NBP2)
  AMOV2=BRANCHP(J+8*NBP2)
  AMGV1=BRANCHP(J+5*NBP2)
```


1.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
10
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

```
      CALL TTL(ML,NL,BLEG BRANCHP,NBP2,ILEP,PQL,NPQ)
C     DO 887 I=1,ML
C887  WRITE(6,999)(BLEG(I+(J-1)*NL),J=1,16)
      IF(IERROR.GT.0)RETURN
      DO 20 IQ=1,ML
      G=BLEG(IQ+5*NL)
      Q=ABS(BLEG(IQ+2*NL))
   20 BLEG(IQ+5*NL)=Q/G
C     BUILD THE CALC1 ARRAY
      DO 21 K=1,ML
      I=ILEP(K)
      J=ILEP(K+NL)
      L=IDIAG(I)
      LM=IDIAG(J)
      CALC1(I+(L-1)*NL)=CALC1(I+(L-1)*NL)+BLEG(K+5*NL)
      CALC1(J+(LM-1)*NL)=CALC1(J+(LM-1)*NL)+BLEG(K+5*NL)
      L=JNEG(K)
      LM=INEG(K)
      IF (L.NE.0)CALC1(I+(L-1)*NL)=CALC1(I+(L-1)*NL)-BLEG(K+5*NL)
   21 IF (LM.NE.0)CALC1(J+(LM-1)*NL)=CALC1(J+(LM-1)*NL)-BLEG(K+5*NL)
C     BUILD CALC2 ARRAY
      N1=N
      IF(NPQL2(N+1).NE.0)N1=N+1
      DO 22 I=1,N
      I1=NPQL2(I)
      CALC1(I1)=1.
   22 CALC2(I1)=PQL(I1)-PQL(I1+NPQ)
      DO 23 I=1,JEM
   23 CALC2(I)=CALC2(I)+PQL(I+NPQ)
      DO 24 I=1,ML
      IF(BLEG(I+4*NL).EQ.0.)GO TO 24
      CALC2(ILEP(I))=CALC2(ILEP(I))-BLEG(I+4*NL)*BLEG(I+5*NL)
      CALC2(ILEP(I+NL))=CALC2(ILEP(I+NL))+BLEG(I+4*NL)*BLEG(I+5*NL)
   24 CONTINUE
      IF(NPQL2(N+1).EQ.0)GO TO 25
      J=NPQL2(N+1)
      CALC2(J)=.001
   25 CALL SIMULT(JEM,ITER,CALC1,CALC2,JCOL,JRENT,JCENT,ICENT,
     1IRENT,IORDER,ICOL,NPQ)
      DO 26 IM=1,JEM
      PQL(IM)=CALC1(IM)
   26 CONTINUE
C     CALCULATE NEW FLOW RATES
      DO 27 IT=1,ML
      IU=ILEP(IT)
      IV=ILEP(IT+NL)
      BLEG(IT+6*NL)=(PQL(IU)+BLEG(IT+4*NL)-PQL(IV))*BLEG(IT+5*NL)
   27 CONTINUE
C     TEST NEW FLOW RATES
      DO 31 IJ=1,ML
      Q=BLEG(IJ+2*NL)
      QNEW=BLEG(IJ+6*NL)
      IF(ABS(Q-QNEW).LE.TOL)GO TO 31
      IF(ABS(Q).GT.1.)GO TO 28
      IF(ABS(QNEW).LE.1.)GO TO 35
```

8.12.4 (Continued)

```
   28 IF(ABS(Q).GT.ABS(QNEW))GO TO 29
      QBIG=QNEW
      GO TO 30
   29 QBIG=Q
   30 IF(ABS((Q-QNEW)/QBIG).GT.TOL)GO TO 35
   31 CONTINUE
      IF(ITER1.GE.1)GO TO 32
      ITER1=ITER1+1
      GO TO 35
   32 CONTINUE
      DO 33 IZ=1,ML
      Q=BLEG(IZ+2*NL)
      IF(Q.EQ.0.)BLEG(IZ+2*NL)=.00001
   33 CONTINUE
      IF(ITER.GE.MAXITER)WRITE(6,34)
      IF(ITER.GE.MAXITER)STOP
   34 FORMAT(10X,42HEXCEEDED MAX NO OF ITERATIONS-CHECK SYSTEM)
      IC=0
      CALL FLOCHEK(IC,BRANCHP,NBP2,BLEG,NL,PQL)
      CALL VCHEK(ML,IC,BRANCHP,NBP2,BLEG,NL,ILEP,PQL,NPQ)
      CALL ACTCHEK(N,IC,BRANCHP,NBP2,PQL,NPQ)
      CALL MTRCHK(BRANCHP,NBP2,IC)
      IF(IC.NE.0)GO TO 1
      RETURN
C     RECALCULATE FLOW RATES
   35 DO 36 I=1,ML
      Q=BLEG(I+2*NL)
      QNEW=BLEG(I+6*NL)
      BLEG(I+2*NL)=(Q+QNEW)/2.
      IF(Q.EQ.0.)BLEG(I+2*NL)=.00001
   36 CONTINUE
      IF(ITER.EQ.MAXITER)GO TO 32
      ITER=ITER+1
      GO TO 15
      END
```

SECTION IX

REFERENCES

1. Micheals, D.G., "MUFAN, A Computer Program for the Analysis of Multi-Loop Fluid Flow Systems, "Aerojet-General Corporation, Technical Memorandum 7975:70-644, June 1971.

2. USAEC-TR-6630, "Handbook of Hydraulic Resistance," Translated from Russian and Printed in Jerusalem by S. Monson, 1966.

3. Technical Paper No. 410, "Flow of Fluids through Valves, Fittings, and Pipe," Crane Co., Chicago, 1957

4. Keller, G.R., "Hydraulic System Analysis," Editors of Hydraulics and Pneumatics Magazine, 1970.

5. Streeter, Victor L., "Fluid Mechanics," McGraw-Hill, 1951.

6. Cambel, A.B., Jennings, B.H., "Gas Dynamics," McGraw-Hill, 1958.

7. Merritt, Herbert E., "Hydraulic Control Systems," John Wiley and Sons, Inc. 1967.

8. Lewis, Ernest E. and Stern, H., "Design of Hydraulic Control Systems," McGraw-Hill, 1962.

9. Report GH20-0205-4, "System/360 Scientific Subroutine Package - Version III, Programmer's Manual," IBM Corp, August 1970.

PRECEDING PAGE BLANK-NOT FILMED

APPENDIX A

26 September 1974
Rev 13 February 1975
Rev 16 July 1979

CALC SUBROUTINE MATHEMATICAL DEVELOPMENT

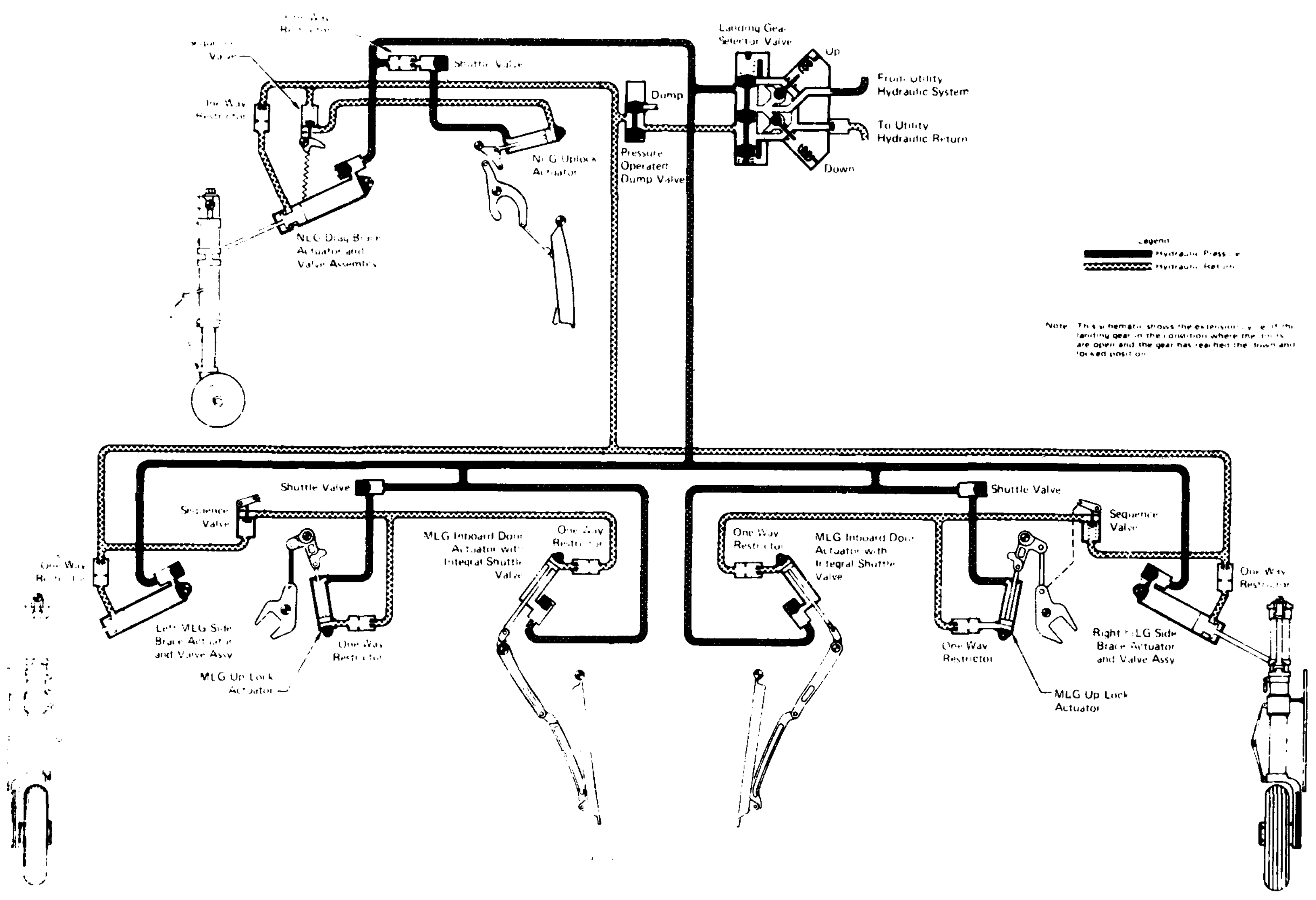
by

R. J. Levek
R. E. Young
P. C. McAvoy

1.0 INTRODUCTION

The creation of a mathematical model to define a physical system requires the establishment of a set of relationships among the individual elements of the system. Operating conditions or parameters are inserted into the system model which responds with solutions for these input values. The actual system often may not be described by textbook relationships, for very few systems actually model the book. A real-world solution must be found that involves solving variables sometimes in terms of these same variables. This type of solution procedure is ideally suited for the iterative processes of the computer.

The mathematical system model developed for SSFAN solves for pressures and flows in a multiple-loop aircraft hydraulic system. Figure A-1 is a schematic of an aircraft landing gear subsystem that is typically solved by the math model. Figure A-2 is a line schematic of Figure A-1 and illustrates the multiple-loop complexity (note cross branching) that is relatively common in aircraft hydraulic systems. Other areas of concern in the development of a good system mathematical model are developing good element models for the pumps, actuators, accumulators, reservoirs, and other elements that have variable flow vs pressure drop characteristics. In a simple unbalanced actuator, the flow and pressure gradients are calculated through area ratios and external force considerations. A math model may handle the flow discontinuity well, but not the pressure one. A variable delivery pump presents many parameters to consider, along with flow in and out of the ports. These parameters include the internal leakage of the pump and the variation of output flow with pump case and inlet pressure, including the decrease of outlet flow when the pump inlet is cavitating.


Landing Gear Selector Valve
Up
Down
From Utility Hydraulic System
To Utility Hydraulic Return
Dump
Pressure Operated Dump Valve
Shuttle Valve
One Way Restrictor
NLG Uplock Actuator
NLG Diag Brace Actuator and Valve Assembly
Shuttle Valve
Sequence Valve
One Way Restrictor
Left MLG Side Brace Actuator and Valve Assy
MLG Up Lock Actuator
MLG Inboard Door Actuator with Integral Shuttle Valve
Right MLG Side Brace Actuator and Valve Assy
Legend
Hydraulic Pressure
Hydraulic Return

Note: This schematic shows the extension cycle of the landing gear in the condition where the doors are open and the gear has reached the down and locked position.

FIGURE A-1
AIRCRAFT LANDING GEAR HYDRAULIC SUBSYSTEM SCHEMATIC

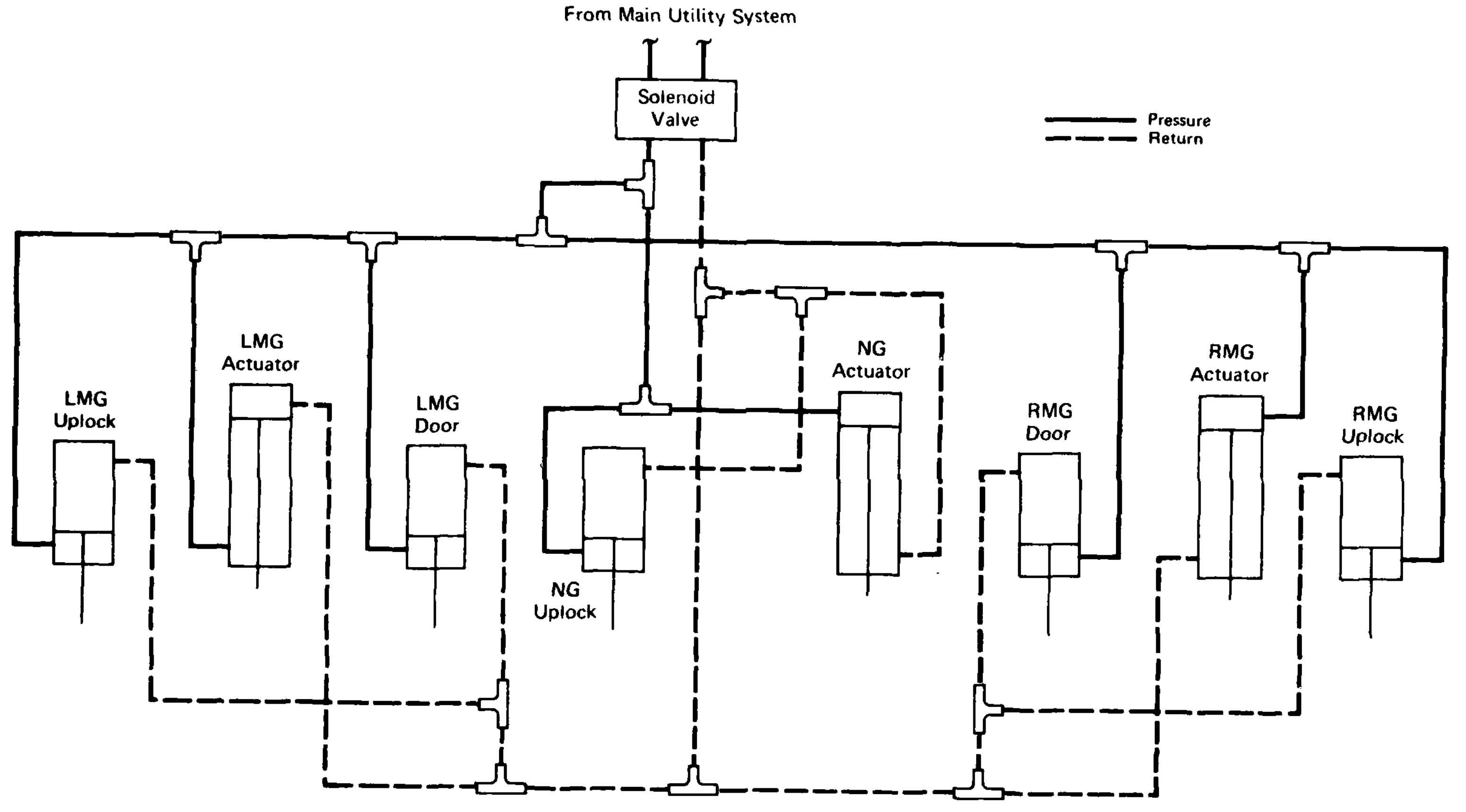


FIGURE A-2
AIRCRAFT LANDING GEAR HYDRAULIC SUBSYSTEM
SIMPLIFIED SCHEMATIC

2.0 SOLUTION METHODS

Two analytical techniques were evaluted for solution of the multiple-loop flow problem. One is a convergence method which balances pressure drops in simple loops. The other is an iterative matrix method which balances flows at branch points.

2.1 Convergence Method

The convergence method compares two legs at a time and relies on prior calculation of values to arrive at new values. A flow division is assumed at branches. Individual element pressure drops along one leg are summed and compared with the pressure drops of the other leg. The flows are changed and new comparisons of pressure drops are made. The iteration continues until the pressures balance. The convergence routine starts at the pump and assumes an exact flow split ($Q_o = Q_1 + Q_2$) of flow combining at each branch point in the system. The initial pump flow is estimated. The convergence is begun at the innermost loop in the system and iterated outwardly. Convergence is accomplished (see Figure A-3) using the initial flow estimate split at P_1 and changing the flows in legs (1) and (2) until the pressure drops in legs (1) and (2) balance at P_2. The next loop to balance is (P_1 to P_4). Using the initial flow split of legs (3) and (5), the pressure drops are compared from (P_3 to P_4). If these do not balance, a new flow split is assumed. When this occurs, the (P_1 to P_3) loop has to be rebalanced. Each outer loop change in flow requires a rebalance of all inner loops. Figure A-4 represents an example of the difficulty involved in applying this method to a simple system. Two legs cannot be paired before one branches in with another leg. Since each loop is balanced separately, redundant computations are required; therefore, this procedure becomes time consuming. (Normally with this technique, one loop will converge in about 7 iterations. For a complex system that contains 20 loops, 7^{20} or approximately 8×10^{16} iterations are required.) This method uses a Newton-Raphson with an Atkins delta squared technique for convergence.

2.2 Matrix Method

The matrix method balances flows for all the loops at one time through the temporary assumption of linear resistance for each leg of the system. The flow rate in each leg is changed through comparison of the calculated flow using the linear resistance and matrix solution pressures at each end with the actual resistance using this calculated flow. Iteration continues until the flows balance at all branch points.

Balancing system flows offers much more flexibility than the convergence method. Each leg in the system is treated independently, allowing one to set up a system of equations to solve for all the leg flows at once. The problem of pressure discontinuities may be handled readily by insertion of the additional pressure rise or drop to the specific legs where these step changes in pressure apply.

D. G. Michaels used this technique in solution of multiple-loop flow problems of low pressure systems. (Reference(A-1)). The transition of this technique to an aircraft high-pressure closed-loop hydraulic system presents many problems. The simple unbalanced area actuator represents a flow and pressure discontinuity. In this case, the pressure-in is converted to a force

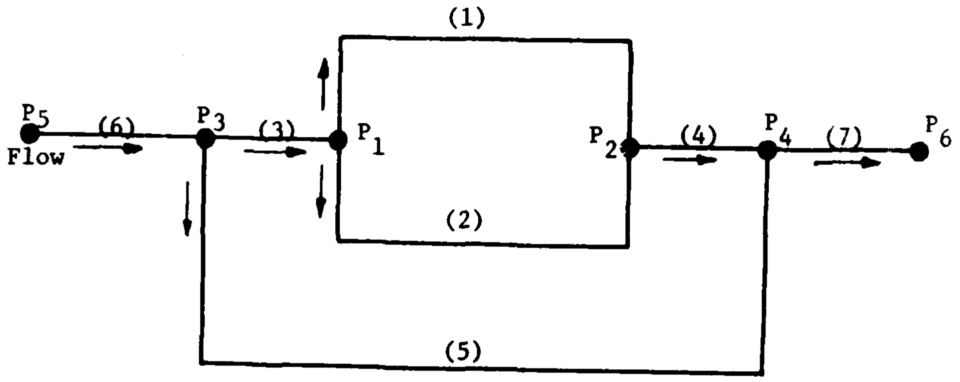


FIGURE A-3

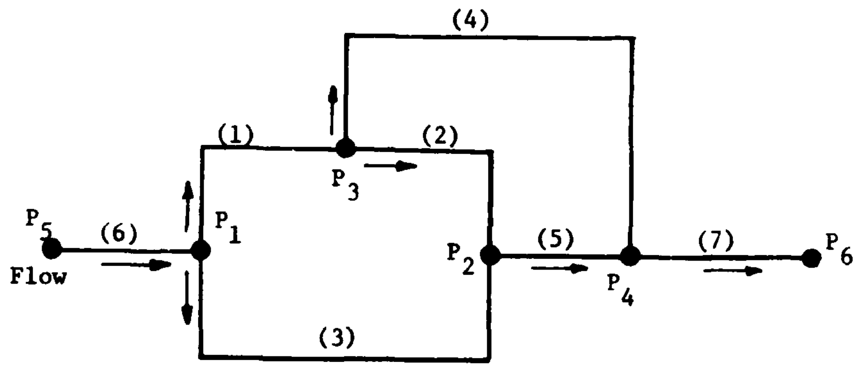
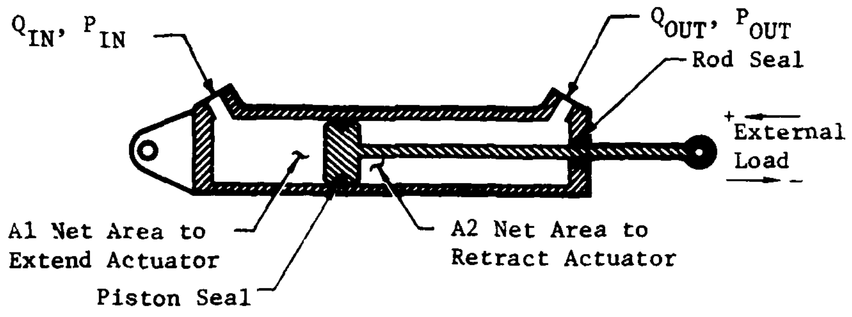
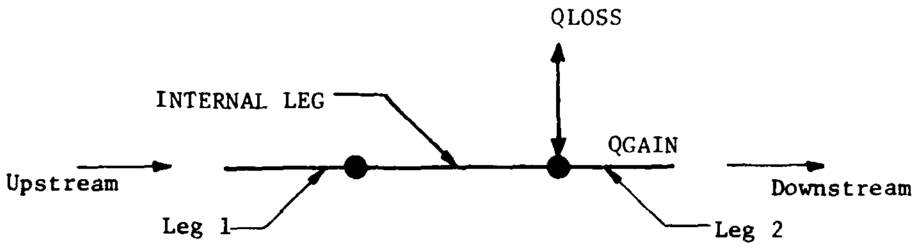


FIGURE A-4

to move the load. The flow-out is proportional to the flow-in, but does not equal it. The pressure-out is a function of the pressure-in, the internal friction, and the actuator external load. When the actuator stalls or in some conditions reverses itself as a result of flight loads, the flow conditions change. These problems were overcome by considering the actuator to be a 2 branch point model. A flow gain or loss term is added, based on the direction of motion of the piston and the ratio of the extend and retract areas of the piston. The associated pressure gradient is applied to the internal leg of the actuator. Figure A-5 compares an actuator to its SSFAN subroutine model equivalent.



ACTUATOR EXTENDING



BRANCH POINT REPRESENTATION

FIGURE A-5

The pressures at the branch points represent the pressures on each side of the actuator piston. The Q_{Loss} term is the difference between Q_{in} and Q_{out}. The pressure rise or loss across the actuator is inserted in the internal leg of the actuator. Other problem areas in the flow balance method included variable delivery pumps, an altitude-pressure dependent reservoir, multiple accumulators, check valves, servo-actuators, relief valves, and one-way restrictors.

The steady-state flow problem of multiple-loop hydraulic systems by definition requires the variables at any instant of time to be directly solvable. Solutions to these variables are time independent, but may reflect any degree of mathematical complexity.

The solution of flows in a hydraulic system requires a knowledge of the pressure drop characteristics of the components in the system. The key to the solution of the flow problem using the matrix technique lies in instantaneously linearizing these characteristics and the solving for a leg resistance factor. Once these factors are known, linear equations for flow may be written at the points where two or more flows meet. A system of n equations and n unknowns is the result. Each term contains only one unknown to the first power. For more than three unknowns, this defines a hyperplane on which the solutions to the system lie. These n values, when substituted back into the n equations, satisfy all of them simultaneously.

2.2.1 Solution Technique

The actual solution development requires the application of Bernouli's equation and the equation of fluid flow continuity to arrive at a resistance factor (see Ref. (A1) and Section 3-2). Conductance is defined as the inverse of resistance, and the conductance of a leg times the pressure difference between the two leg ends yields a flow. With these basic facts, an application of the flow continuity equation to any multiple loop system will yield a system of simultaneous linear equations. For illustration, the simple system in Figure A-6 is developed.

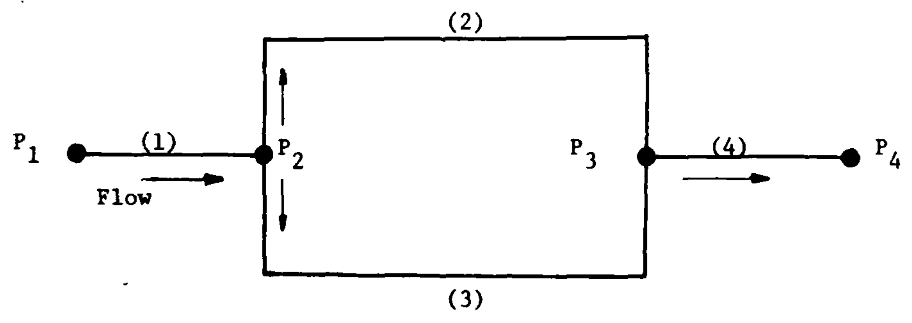


FIGURE A-6

One can write an equation for each leg in the system based on conductance, pressure drop, and flow. For Figure A-6, these are:

$$\begin{aligned}
&\text{Leg 1} \quad Q_1 = G_1(P_1-P_2)\\
&\text{Leg 2} \quad Q_2 = G_2(P_2-P_3)\\
&\text{Leg 3} \quad Q_3 = G_3(P_2-P_3)\\
&\text{Leg 4} \quad Q_4 = G_4(P_3-P_4)
\end{aligned} \tag{1}$$

Where Q = leg flow
G = leg conductance
P = pressure at the corresponding points

Applying the continuity equation in terms of flow to each pressure point in the system, one may write the following equations:

$$\begin{aligned}
&G_1(P_1-P_2) = 0\\
&G_1(P_2-P_1) + G_2(P_2-P_3) + G_3(P_2-P_3) = 0\\
&G_2(P_3-P_2) + G_3(P_3-P_2) + G_4(P_3-P_4) = 0\\
&G_4(P_4-P_3) = 0
\end{aligned} \tag{2}$$

Rewriting in terms of pressure:

$$\begin{aligned}
&P_1\ (+G_1)+ P_2(-G_1) = 0\\
&P_1(-G_1) + P_2(G_1+G_2+G_3) + P_3(-G_2-G_3) = 0\\
&P_2(-G_2-G_3) + P_3(G_2+G_3+G_4) + P_4(-G_4) = 0\\
&P_3(-G_4) + P_4(G_4) = 0
\end{aligned} \tag{3}$$

Writing the set of equations (3) in matrix form:

$$\begin{bmatrix} G_1 & -G_1 & 0 & 0 \\ -G_1 & G_1+G_2+G_3 & -G_2-G_3 & 0 \\ 0 & -G_2-G_3 & G_2+G_3+G_4 & -G_4 \\ 0 & 0 & -G_4 & G_4 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

Examining the above G matrix, one notes that the sum of the elements in any column equals zero. This reflects the conditions of continuity imposed on each pressure point. One also may observe the symmetry of the G matrix. Evaluating the determinant of G yields a value of zero. The G matrix is singular. Singular systems have their application in eigenvalue problems and consequently no unique solution exists for this set of equations. When adequate boundary conditions to the system are specified, the singularity of the G matrix will be removed and the system may be solved. Before boundary conditions are imposed, note that the diagonal elements are all positive. This follows the convention that resistive forces act in a positive way to oppose flow.

One must select reasonable physical constraints for an operating aircraft hydraulic system. Assume point 1 in Figure A-6 to be the pressure port of a variable delivery pump and point 4 the pressure in a bootstrap reservoir referenced to pump out pressure. For simplicity a constant pressure of 3000 psi at point 1 and 50 psi at point 4 are chosen. Since P_1 and P_4 are now known, a new set of equations describing the system may be written.

Branch point equations for any system may now be written

where

$$\sum_L G_L \left[P_I - P_J \pm \Delta P_L\right] - \sum_L \pm Q_L = 0 \qquad (5)$$

P_I	= Pressure at branch point I
P_J	= Pressure at branch point J
Q_L	= Fixed flow rate in leg L
G_L	= Fluid conductance in leg L
ΔP_L	= Fixed pressure change in leg L

For the sample system of Figure A-6, these equations are:

$$\begin{bmatrix} G_1 & -G_1 & 0 & 0 \\ -G_1 & G_1+G_2+G_3 & -G_2-G_3 & 0 \\ 0 & -G_2-G_3 & G_2+G_3+G_4 & -G_4 \\ 0 & 0 & -G_4 & G_4 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} -Q_1-\Delta P_1 G_1+\Delta P_2 G_2 \\ -Q_2-\Delta P_2 G_2+\Delta P_3 G_3 \\ Q_3 \ -\Delta P_2 G_2+\Delta P_3 G_3 \\ Q_4 \ -\Delta P_3 G_3+\Delta P_4 G_4 \end{bmatrix} \qquad (6)$$

Since P_1 and P_4 are known, the equations for branch points 1 and 4 are replaced with the constant pressure equations:

$$P_1 = 3000 \qquad (7)$$
$$P_4 = 50$$

With these equations replaced, the final array appears in the form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -G_1 & G_1+G_2+G_3 & -G_2-G_3 & 0 \\ 0 & -G_2-G_3 & G_2+G_3+G_4 & -G_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} 3000 \\ -Q_2-\Delta P_2G_2+\Delta P_3G_3 \\ Q_3-\Delta P_2G_2+\Delta P_3G_3 \\ 50 \end{bmatrix} \quad (8)$$

Examining equation (9), one may note how fixed pressure rises or drops, or constant flows are accounted for in this method. For a fixed pressure rise in Leg L , ΔP_L is included in the summing of conductances around a branch point. Constant flows, Q_L are also summed around the specified branch point I.

The sign used for Q_L depends on the flow direction only. The sign for ΔP_L depends on the flow direction which in turn defines whether the term is a pressure rise or drop. If flow is being added to a system at branch point I, then the direction of Q_L is positive. For a pressure drop in a leg, the sign of ΔP_L is negative for an upstream branch point and positive for the downstream branch point. Applying equation (5) to every branch point, one may write the total G matrix equation for a system of branch points in this form:

$$GP = K \quad (9)$$

Where:

G = matrix of conductance coefficients

P = unknown branch point pressures

K = system constants

For a given coefficient matrix and constant matrix, the problem narrows down to finding a good matrix solution technique.

Many methods are available to solve systems of simultaneous linear equations. Cramer's rule involving determinants is one of them. Briefly, Cramer's rule gives solutions for the variables by evaluating determinants resulting from the constant terms of the system, and constants from the variable terms. For large numbers of equations, the nth determinant is evaluted by developing a row or column and then developing each cofactor in turn. This procedure results in n! multiplications. A system of 15 equations would require 15! or approximately 10^{12} multiplications. Since a computer can do about 10,000 multiplications per second, it would take a little less than 4 years to solve a system of 15 equations. Fortunately other solution methods exist that are simpler and less time consuming. One of these methods is Gauss-Jordan elimination using a compressed matrix format. This method is used in the Calc matrix solution. The Gauss-Jordan solution process (References A2, A3, and A8) is eased on the three elementary row operations:

1. Interchange any two rows
2. Multiplication of a row by a scalar
3. Addition of a multiple of one row to another row.

For a system of linear equations

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdot \quad \cdot + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdot \quad \cdot + a_{2n}x_n = b_2 \\ \cdot \qquad \cdot \qquad \cdot \quad \cdot \qquad \cdot \qquad \cdot \\ \cdot \qquad \cdot \qquad \cdot \quad \cdot \qquad \cdot \qquad \cdot \\ a_{m1}x_1 + a_{m2}x_2 + \cdot \quad \cdot + a_{mn}x_n = b_n \end{array}$$

or, as expressed in matrix form

$$\begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{12} & \cdot & \cdot & a_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \cdot & \cdot & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

The matrix is then searched to find the row with the minimum number of non-zero elements. If two or more rows satisfy this criteria, the row with the smallest index is selected. After the pivotal row has been selected, the columns with non-zero entries are searched and the column with the minimum number of non-zero entries is selected. If two or more columns contain the same number of elements, the column with the smallest number of entries is selected.

The intersection of the pivotal row and column define one element in the array, the pivotal element. The pivotal row is then normalized by dividing it by the pivotal element. The rows with elements in the pivotal column are then selected. The pivotal row is multiplied by the negative of the element in the pivotal column of the row being operated on and the two rows are added.

After all the elements in the pivotal column other than the pivotal element are eliminated, a new pivotal element is selected.

After a pivotal element has been selected from each row, a triangular matrix is obtained. (A triangular matrix is a square matrix that has all elements either above or below the main diagonal). The triangularized system appears in matrix form below:

$$\begin{bmatrix} 1 & a'_{12} & \cdot & \cdot & a'_{1n} \\ 0 & 1 & \cdot & \cdot & a'_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ \cdot \\ \cdot \\ b'_n \end{bmatrix}$$

or writing out the equations:

$$\begin{aligned} x_1 + a_{12}\, x_2 + \cdot\cdot\cdot\cdot\cdot\ a_{1n}\ x_n &= b'_1 \\ x_2 + \cdot\cdot\cdot\cdot\cdot\ a_{2n}\ x_n &= b'_2 \\ x_n &= b'_n \end{aligned}$$

Using the last equation which determines X_n, substitute it into the next to the last equation to determine X_{n-1} and so on for the remainder of the equations.

In writing the SSFAN program, the objective was to solve a large, sparse, system and to minimize the computer storage and time required. This was accomplished by using a compressed array technique which stores a minimum number of zero entries. To minimize storage, a pivot selection method that produces the minimum number of non-zero entries during the solution process must be employed. The minimum row-minimum column pivot selector has been shown (reference A8) to be consistently faster and reduce fewer terms than other pivot selectors.

The compressed matrix technique used (reference A8) requires the storing of a minimum of zero valued entries in addition to the non-zero terms. Since any pressure point in the SSFAN program has at most four connections, any equation will have at most five non-zero entries (one for the pressure point and one for each of the four connected pressure points.). As rows in the matrix are added in the solution process, new terms can be generated to provide more than five entries in any one row. Experimenting with large, complex, systems ranging up to 58 pressure points has shown the generation of no row with more than eight non-zero entries. By storing only 8 entries in each row, as opposed to 58, the storing of 50 zero entries per row or 2900 total entries can be eliminated.

The compressed matrix technique generates and uses six system pressure point identification arrays. Specifically, these arrays are:

JCOL:

1) Dimension (M,5)

2) The final JCOL array (in compressed form) identifies the columns in a square CALC1 array which are filled with non-zero terms. The rows of JCOL correspond to the rows of CALC1, and the elements in each row of JCOL correspond to the column number in each row of CALC1.

3) Note: JCOL describes a square CALC1 array in order to be compatible with the solution technique in SIMULT.

IDIAG:

1) Dimension (M)

2) The IDIAG array identifies which columns of CALC1 contain the positive conductance values. IDIAG(1) corresponds to the column in which the positive conductance is located in the first row of the CALC1 array. IDIAG(2) corresponds to the column in which the positive conductance is located in the second row of the CALC1 array.

JNEG:

1) Dimension (ML)

2) The JNEG array identifies which column in CALC1 contains the first appearance of a negative conductance value in the CALC1 array. For example, JNEG(4) records the first negative conductance value of Leg 4. If JNEG(4) = 3, then the first time a negative leg 4 conductance appears is in row 4, column 3 of the CALC1 array.

INEG:

1) Dimension (ML)

2) The INEG array differs from the JNEG array only in one respect, that being the INEG array stores the second appearance of a negative conductance value.

JRENT:

1) Dimension (M)

2) The JRENT array identifies the number of non-zero entries in each row of CALC1.

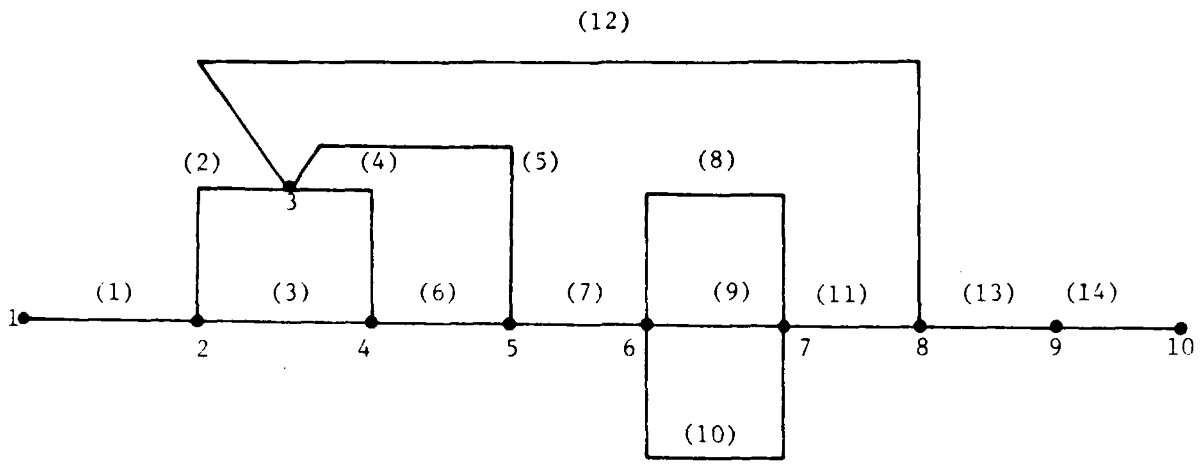
JCENT:

1) Dimension (M)

2) The JCENT array identifies the number of non-zero entries in each column of CALC1.

3) Note: JCENT describes a <u>square</u> CALC1.

To understand how the system pressure point identification arrays are built, the example system of Figure A-7 is developed below.



1	1	2
2	2	3
3	2	4
4	3	4
5	3	5
6	4	5
7	5	6
8	6	7
9	6	7
10	6	7
11	7	8
12	3	8
13	8	9
14	9	10

ILEP ARRAY

NUMBER OF PRESSURE POINTS M = 10

NUMBER OF LEGS ML = 14

PRESSURE POINT 2 IS A CONSTANT PRESSURE POINT

FIGURE A-7

CALC EXAMPLE SYSTEM

I. JCOL is filled with column numbers that contain non-zero entries in a square-CALC1.

```
C      STORE NON-ZERO COLUMN NUMBERS IN JCOL
       DO 1001 K=1,ML
       I=ILEP(K,2)
       J=ILEP(K,3)
       DO 1001 K1=1,2
       J1=I
       IF (K1.EQ.2)J1=J
       DO 1001 K2=1,2
       J2=I
       IF (K2.EQ.2)J2=J
       DO 1001 K3=1,5
       J3=JCOL(J1,K3)
       IF (J3.NE.0)GO TO 1001
       JCOL(J1,K3)=J2
       J2=0
  1001 IF (J3.EQ.J2)J2=0
```

$$JCOL = \begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 2 & 0 & 0 & 0 \\ 2 & 1 & 2 & 3 & 4 & 0 \\ 3 & 2 & 3 & 4 & 5 & 8 \\ 4 & 2 & 4 & 3 & 5 & 0 \\ 5 & 3 & 5 & 4 & 6 & 0 \\ 6 & 5 & 6 & 7 & 0 & 0 \\ 7 & 6 & 7 & 8 & 0 & 0 \\ 8 & 7 & 8 & 3 & 9 & 0 \\ 9 & 8 & 9 & 10 & 0 & 0 \\ 10 & 9 & 10 & 0 & 0 & 0 \\ \hline \end{array}$$

II. Rows with constant pressure points are zeroed and the diagonal element saved.

```
C      LOAD CONSTANT PRESSURE NODES INTO JCOL
       DO 1009 K=1,N
       I1=NPQL2(K)
       DO 1010 J1=1,5
  1010 JCOL(I1,J1)=0
  1009 JCOL(I1,1)=I1
```

$$JCOL = \begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 \\ 3 & 2 & 3 & 4 & 5 & 8 \\ 4 & 2 & 4 & 3 & 5 & 0 \\ 5 & 3 & 5 & 4 & 6 & 0 \\ 6 & 5 & 6 & 7 & 0 & 0 \\ 7 & 6 & 7 & 8 & 0 & 0 \\ 8 & 7 & 8 & 3 & 9 & 0 \\ 9 & 8 & 9 & 10 & 0 & 0 \\ 10 & 9 & 10 & 0 & 0 & 0 \\ \hline \end{array}$$

III. JRENT and JCENT are built. JCOL is rearranged so its entries are in increasing order, and IDIAG is then built.

```
C     BUILD JRENT,JCENT; REORDER JCOL; BUILD IDIAG
      DO 1002 K=1,M
      KOUNT=0
      DO 1011 K8=1,5
      DO 1011 K9=1,M
 1011 IF (JCOL(K9,K8).EQ.K)KOUNT=KOUNT+1
      JCENT(K)=KOUNT
      KOUNT=0
      DO 1003 K1=1,5
 1003 IF (JCOL(K,K1).NE.0)KOUNT=KOUNT+1
      JRENT(K)=KOUNT
      DO 1004 K2=1,KOUNT
 1004 D(K2)=JCOL(K,K2)
      DO 1005 K4=1,KOUNT
      TEST=0
      DO 1006 K5=1,KOUNT
      IF (D(K5).LT.TEST)GO TO 1006
      TEST=D(K5)
      TEST2=K5
 1006 CONTINUE
      K6=KOUNT+1-K4
      JCOL(K,K6)=TEST
 1005 D(TEST2)=0
      K7=JRENT(K)
      DO 1002 KOUNT=1,K7
 1002 IF (JCOL(K,KOUNT).EQ.K)IDIAG(K)=KOUNT
```

JCENT =

1	4	4	3	4	3	3	4	3	2

JRENT =

2	1	5	4	4	3	3	4	3	2

$$JCOL = \begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 \\ 3 & 2 & 3 & 4 & 5 & 8 \\ 4 & 2 & 3 & 4 & 5 & 0 \\ 5 & 3 & 4 & 5 & 6 & 0 \\ 6 & 5 & 6 & 7 & 0 & 0 \\ 7 & 6 & 7 & 8 & 0 & 0 \\ 8 & 3 & 7 & 8 & 9 & 0 \\ 9 & 8 & 9 & 10 & 0 & 0 \\ 10 & 9 & 10 & 0 & 0 & 0 \\ \hline \end{array}$$

IDIAG =

1	1	2	3	3	2	2	3	2	2

IV. JNEG and INEG are built to record locations of off diagonal elements.

```
C     BUILD INEG,JNEG
      DO 1007 K=1,ML
      I=ILEP(K,2)
      J=ILEP(K,3)
      I1=JRENT(I)
      J1=JRENT(J)
      INEG(K)=0
      JNEG(K)=0
      DO 1008 KOUNT=1,I1
 1008 IF (JCOL(I,KOUNT).EQ.J)JNEG(K)=KOUNT
      DO 1007 KOUNT=1,J1
 1007 IF (JCOL(J,KOUNT).EQ.I)INEG(K)=KOUNT
```

JNEG =

2	0	0	3	4	4	4	3	3	3	3	5	4	3

INEG =

0	1	1	2	1	2	1	1	1	1	2	1	1	1

The JNEG, INEG, and IDIAG arrays are used in CALC to build the compressed CALC1 array. The compressed CALC1 array is a square CALC array with the non-zero terms compressed together and the zero entries removed. The JCOL array is passed to SIMULT to be used in the solution of the system. The JCOL array is copied into ICOL in the SIMULT routine with ICOL being updated each time a row operation is performed. JRENT and JCENT are also transferred to SIMULT to be used in the selection of pivotal elements.

For large systems of equations, considerable roundoff errors may occur in the solution matrix using a digital computer. In SIMULT, operations known to result in a zero or one are bypassed and the element set to the correct value. Also by using minimum row-minimum column pivoting, significantly fewer eliminations are performed. Roundoff error is roughly dependent on the square root of the number of operations involved in the elimination.

Another area of concern in the solution of simultaneous linear equations is truncation errors. By the use of double precision, more accuracy may be obtained, but computation time and computer cost become greater factors. In SSFAN the pressure drops are computed based on a flow and the resistance in each leg. CALC computes the conductance from the SSFAN calculated pressure drops. The accuracy of conductance is therefore dependent on how well one modeled the element flow vs pressure drop characteristics. These characteristics are assembled linearly for each element in a leg: then, given a flow, a conductance term results. Refer to Section 6 for a further discussion of this concept.

2.2.2 Solution Summary and Convergence Criteria

Applying Gauss-Jordan elimination with minimum row-minimum column pivoting to the SSFAN hydraulic system math model requires an iteration technique. The physical problem to be solved involves non-linear values of system resistances. A unique solution exists for any hydraulic system. To find this solution, one first initializes the flows in the system. The flow estimates are substituted into the equations that represent the steady-state pressure drops in the legs. The conductance value is then computed as G = Q/DP. This procedure is followed for every leg. The values or leg conductances are now substituted into the G matrix, which, combined with the system constants, are solved through Gauss-Jordan elimination for pressures at each point in the system. A new pressure drop for each leg is now computed using the new calculated values of pressure, and with previous values of leg conductance, a new flow estimate is made. The procedure is repeated until the old flow estimate and the new flow estimate are within a specified tolerance for all legs in the system. The new flow estimate used to update the iteration makes this method converge within a reasonable number of iterations.

The new calculated flow from the matrix solution is averaged with the old flow guess to yield another flow guess (Q') or

$$Q' = \frac{Qnew + Qold}{2} \tag{30}$$

From a computation viewpoint, one may readily agree to the simplicity of this procedure. No prior values of flow are required; only the corrected value. There are faster root finding convergence methods such as Newton's or Atkin's, but these may also diverge under certain conditions. Reference A6 states that the method of bisection or interval halving technique is "guaranteed to locate a root of the equation". Reference A7 states that "The greatest virtue of the bisection method is that it is virtually assured to converge a root". A few graphic examples will serve to illustrate the validity of this procedure.

Consider the solution of the flows and pressures of the system in Figure A-6. A pressure drop relation for each leg in the system may be written as:

$$DP(I) = K_1 + K_2Q(I) + K_3Q(I)^{1.75} + K_4Q\ (I)^2 \tag{16}$$

where:

$DP(I)$ = Pressure drop in Leg I

$Q(I)$ = Flow in Leg I

K_1, K_2, K_3, K_4 = Leg resistance coefficients for laminar, transition or turbulent flow

Applying Equation (16) to Leg (2) of Figure A-6, a graph of pressure drop vs flow may be plotted (Figure A-8). The unique solution pressure drop in Leg (2) is constant, hence the straight line ΔP_{12} is drawn. Based on the old flow estimate (Q_o), a new flow estimate (Q_1) is chosen to correspond to ΔP_{12}. This is shown graphically by drawing a resistance line R_o through the origin and finding the point where R_o crosses the ΔP_{12} line. This process is repeated to yield a Q_2 and so on.

One notes that after a few iterations, the flows start diverging. Using the correction Equation (15), a better flow guess and step size can be made.

In Figure A-9, Q1 is calculated in the same manner as in Figure A-8, the difference now being that instead of using Q1 for generating a resistance slope, an average of Q_o and Q1 (Q_{o1}) is used. One can see from Figure A-8 that this procedure converges rapidly to a solution, not only for Leg (2), but every leg in the system.

Briefly, the solution method for an entire system will follow the procedure below.

(1) An initial estimate is made for all the flows in the system, as was done for leg (2) in Figure A-9. This yields a specific calculated ΔP for each leg.

(2) Knowing this ΔP and the estimated flow, calculate a conductance factor for each leg.

(3) Insert these factors and system constants (as ΔP_{12} in Figure A-8) in equation (10).

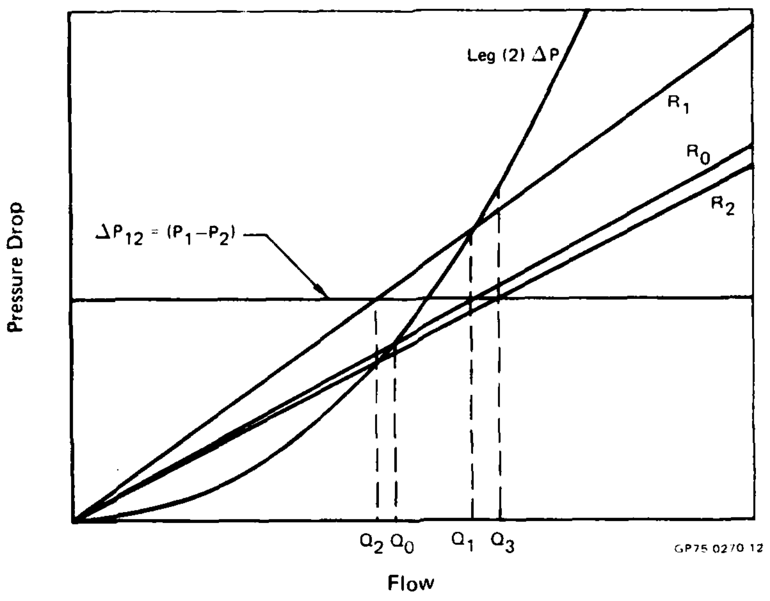


FIGURE A-8
PRESSURE DROP vs FLOW
Leg (2)

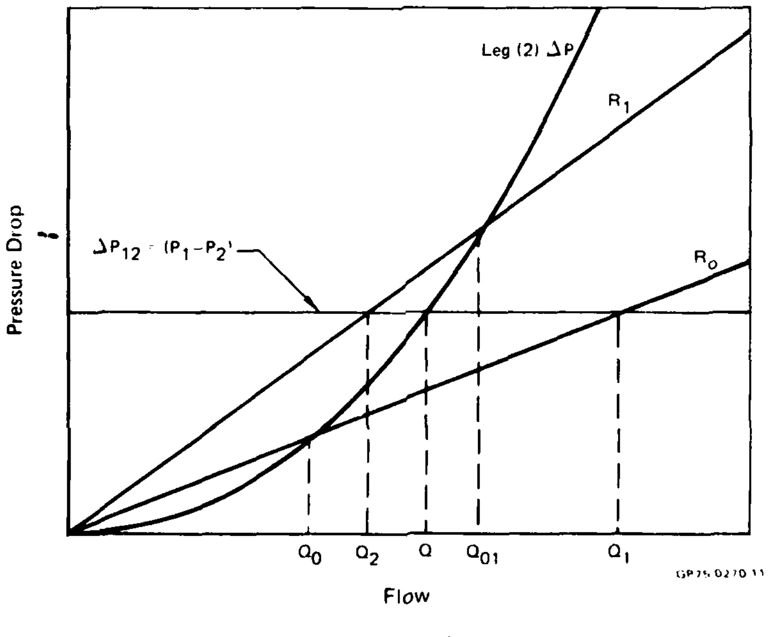


FIGURE A-9
PRESSURE DROP vs FLOW
Leg (2)

(4) Solve for point pressures and using the previous calculated conductance factors, claculate a new flow (Q_1 in Figure A-8).

(5) Test the new flow for convergence, using different convergence criteria for flows greater than one GPM than for flows less than one GPM.

(6) If these convergence criteria are not met, compute a corrected flow (as Q_{01} in Figure A-9) for each leg and go to Step (1) and repeat, using the flow from Step (6). The flows will converge to satisfy each leg's characteristic flow pressure drop curve and constant characteristics.

Different methods have been tried to weight and the corrected solution in the direction of the new flow estimate, but none works as well as this interval halving technique. An interval is found that contains the final solution pressure drop value, and repeatedly halved until the solution is found.

The rate of convergence of the interval-halving method is dependent on the way each new approximation is made. The interval within which the flow will be located after i iterations is $(1/2)^i$ times the original interval, Reference A5. Thus, the error in the ith approximation may be not more than $(1/2)^i$ times the initial interval on Q. For this method to converge, the pressure drop equation (16) must be single valued for one iteration and only one value of flow should exist in the initial interval (Reference 4). The solution is slightly starting value dependent but for the range of flows for aircraft hydraulic systems an initialization of 10 gpm has been included in SSFAN. Flow initializations of less than one and greater than 100 gpm have been tried, resulting in 3 to 5 more iterations before convergence.

One may look at the entire mathematical model of the steady state flow system as being divided into two major parts. First a calculation of the pressure drops in the system based on computer estimated flows is made, then pressure points in the system are calculated based on the previous pressure drops which give the conductance values. The more accurate the pressure point values from the Gauss-Jordan elimination method with pivoting the more accurate the flow guesses will be, and the better the solutions. Thus propagation error does play an important role in this procedure. A mistake in one selection of a new flow may have its effect on the final calculated flows in the system as well as system pressures. A $\pm.1$ allowable error tolerance in flow can mean a ±10 psi difference in pressures. That is why an error tolerance of $\pm.001$ gpm is used when comparing old and new flows for final convergence.

The SSFAN model was developed to solve any multiple-loop aircraft hydraulic system. This study has briefly surveyed the methods available for this solution and the thought behind the selecting of certain procedures over others. Also, the methods chosen were given a short summary to present their main attributes and how SSFAN overcame some of their weaknesses. The solution procedure for writing equations was presented and the method used in solving this system of equations was illustrated. Convergence techniques were discussed along with the part computational errors played in this iteration process.

The development of any mathematical model for a sophisticated system as an aircraft hydraulic system, requires much time and effort to arrive at a realistic portrayal of the actual system. The model once verified through actual performance test data must be flexible enough to meet the requirements for any new system the engineer may study.

A3.0 REFERENCES

A1. Micheals, D. G., "MUFAN, A Computer Program for the Analysis of Multi-Loop Fluid Flow Systems," Aerojet-General Corporation, Technical Memorandum 7975:70-644, 21 June 1971.

A2. Dorn, W. S. and McCracken, D. D., "Numerical Methods and FORTRAN Programming," John Wiley and Sons, Inc., 1964.

A3. Kelly, L. G., "Handbook of Numerical Methods and Applications," Addison-Wesley Publishing Company, 1967.

A4. Beckett, R. and Hurt, J., "Numerical Calculations and Algorithms," McGraw-Hill, Inc., 1967.

A5. Grove, Wendell E., "Brief Numerical Methods," Prentice Hall, 1966.

A6. McCalla, T. R., "Introduction to Numerical Methods and Fortran Programming", 1967.

A7. Pennington, R. H., "Introductory Computer Methods and Numerical Analysis," McMillan, 1965.

A8. Key, J.E., "Computer Program for Solution of Large, Sparse, Unsymmetric Systems of Linear Equations," International Journal for Numerical Methods in Engineering, Vol. 6, 1973.

APPENDIX B

FLUID VISCOSITY AND FLUID VISCOSITY CORRECTION WITH PRESSURE

BY

Robert E. Young

Date 10 August 1974
Rev. 10 April 1975

BASED ON
AFML TR-66-107 PART I
FLUIDS, LUBRICANTS, FUELS
AND RELATED MATERIALS
AND
ASTM DESIGNATION D341-74
VISCOSITY-TEMPERATURE
CHARTS FOR LIQUID
PETROLEUM PRODUCTS

APPENDIX B

Fluid Properties

Viscosity is one of the most important properties of the fluid from the standpoint of the steady state analysis. Viscosity is the internal resistance to movement of one portion of a liquid in relation to another. Both temperature and pressure affect viscosity and are taken into account in the SSFAN program.

Viscosity is defined by Newton's Law which states that at a given point in a fluid, the shearing stress is proportional to the rate of shear.

$S = \eta R$	S - shearing stress
	R - rate of shear
	η - coefficient of viscosity

Fluids which flow in accordance with Newton's Law are called "Newtonian Fluids" where the coefficient of viscosity remains constant as the flow rate or rate of shear is varied. This applies to lamirar flow only.

As the flow rate is increased, the ratio of the shearing stress to rate of shear decreases greatly at a critical point where the fluid flow is turbulent. The fluid no longer behaves in Newtonian fashion, therefore other relationships for the coefficient of viscosity have to be utilized.

A typical Non-Newtonian fluid is MIL-H-5606 fluid commonly used in Military Aircraft. This is a hydrocarbon fluid derived from petroleum which with additives exhibits, (1) high viscosity index to give operability over a wide temperature range, (2) resistance to oxidation, (3) good wear resisting properties, (4) resistance to rust and to foaming, and (5) a long operational life.

A typical Newtonian fluid is MIL-H-83282 fluid which has recently had limited usage in Military Aircraft. This fluid is a synthetic hydrocarbon and with additives, exhibits characteristics much the same as MIL-H-5606 except that the viscosity at low temperatures (-40^{C} and below) increases to about five times that of MIL-H-5606.

The Skydrol 500 fluids are another example of a Newtonian fluid. These are based on phosphate esters containing small amounts of additives to give properties required by commercial or transport aircraft. These fluids exhibit fire resistance and were designed to overcome the hazards of flammable hydraulic fluid coming in contact with hot brakes, exhaust manifolds or other ignition sources if a line broke or a leak occurred.

<u>Viscosity-Temperature</u>

The ASTM chart is frequently quoted as being on the Walther log log formula. This statement is incorrect, see Reference (B1), although they are similar in form. MacCoull published a chart in 1921 using the log log relationship plus a constant. This was the form

$$\log \log (\mathrm{cSt} + \mathrm{constant}) = A - B \log T \qquad (1)$$

His study resulted in selecting 0.7 as the constant for publication in the 1927 International Critical Tables. An ASTM committee undertook a study of the chart and selected .8 as the constant for its first chart publication in 1932. Their report credits the relationship to MacCoull. The ASTM committee published an improved chart in 1937, later revised in 1943, in which a constant of .6 was used down to viscosities of 1.5 centistokes. This chart ASTM Designation D341 was used until 1974. In 1974 the standard was revised to ASTM Designation D341-74 to "provide a significant improvement in linearity over charts previously available under method D341-43," see Reference (B3). It is noteworthy that the constant has been revised back to .7 which was MacCoull's original constant in 1927 and his equation may be written

$$\log \log (\mathrm{cSt} + .7) = A - B \log T \qquad (2)$$

The Walther equation was first published in 1928 without the constant. In 1931 a constant of .8 was added. The Walther equation has continued to get extensive use, particularly in Europe. The new ASTM charts of 1974 were derived with computer assistance to provide linearity over a greater range on the basis of the most reliable of modern data. They are set up for temperature in degrees Celsius, with provisions for degrees Rankine during the interim period of changing to the metric system.

The complete design equation for the chart is not useful for inter-calculations of kinematic viscosity and temperature over the full chart kinematic viscosity range. More convenient equations which agree closely with the chart scale are given below, Reference (B2). These are necessary when calculating kinematic viscosities smaller than 2.0 centistokes. It has been demonstrated, that the improved chart permits reliable linear extrapolation into low-viscosity, high temperature regions which were not possible previously. These equations are used for calculating viscosities in SSFAN.

$$\log \log Z = A - B \log T \qquad (3)$$

$$Z = \nu + .7 + \exp(-1.47 - 1.84\nu - .51\nu^2) \qquad (4)$$

$$\nu = [Z - .7] - \exp(-.7487 - 3.295[Z-.7] + .6119[Z-.7]^2 - .3193[Z-.7]^3) \qquad (5)$$

A and B = constants

ν = kinematic Viscosity

T = Temperature °R

Viscosity-Pressure

The viscosity of all fluids changes with pressure. These effects are sometimes neglected in low pressure systems where moderate changes in pressure are involved. However, for better analytical accuracy, these effects should be included in the analyses. Petroleum base fluids show an appreciable increase in viscosity with pressure.

SSFAN uses the following equation to adjust the viscosity at higher pressures:

$$\mu_p = \mu_o \, e^{2.3 \alpha^a P x 10^{-4}} \qquad (6)$$

μ_p = kinematic viscosity (centistokes) at pressure P

μ_o = kinematic viscosity (centistokes) at atmospheric pressure

P = pressure in Psig

e = base of Napierian logarithms = 2.718

α = pressure coefficient of viscosity psig^{-1}

a = $\frac{560}{{}^oR}$ an emperically derived temperature-dependent exponent

Based on the work of Professor E. Klause at Pennsylvania State University, the pressure coefficient of viscosity for a number of fluids were determined. A cross reference is given by Professor Klause between the ASTM slope, see Figure B-1 and the α coefficient.

The slope of the viscosity curve is taken from Figure B-2. If the viscosity of the fluid at two temperatures of a fluid is known, the slope can be established. The viscosity temperature line forms an angle "A" with the lines of constant viscosity. The tangent of "A" is the slope of the centistoke viscosity-temperature line. It is also the vertical distance "Y" along a line of constant temperature in inches divided by the horizontal distance "X" in inches measured along a line of constant viscosity.

where: ASTM Slope = tangent A = $\frac{Y \text{ (in)}}{X \text{ (in)}}$

Slope values would obviously be negative, but are ignored by convention. See Figure B-2 for an example of ASTM slope. The Figure B-1 pressure coefficient is then determined using an atmospheric viscosity at a temperature of 100° F for the slope determined from Figure B-2.

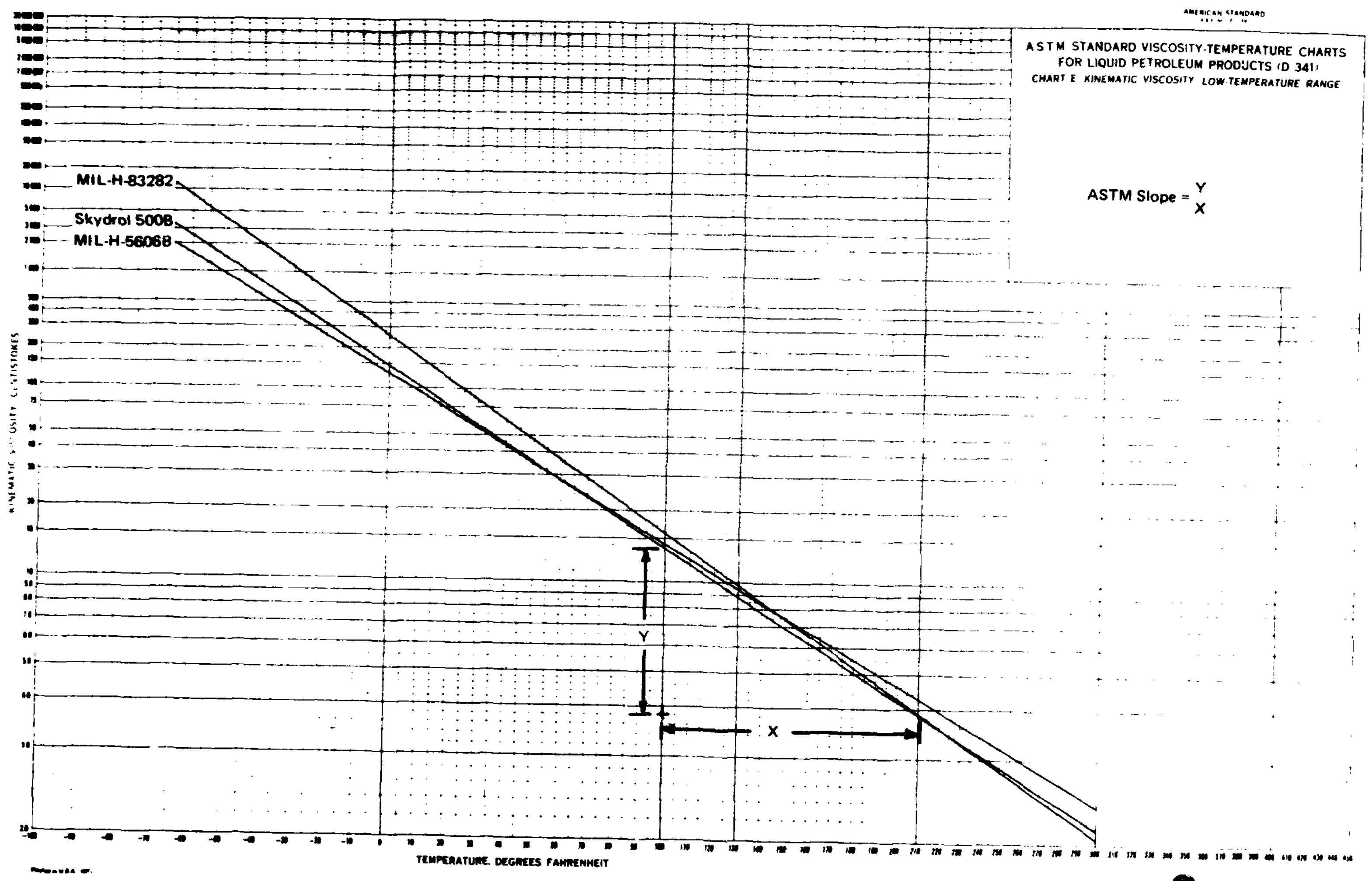


FIGURE B-1
ASTM VISCOSITY-TEMPERATURE CHART

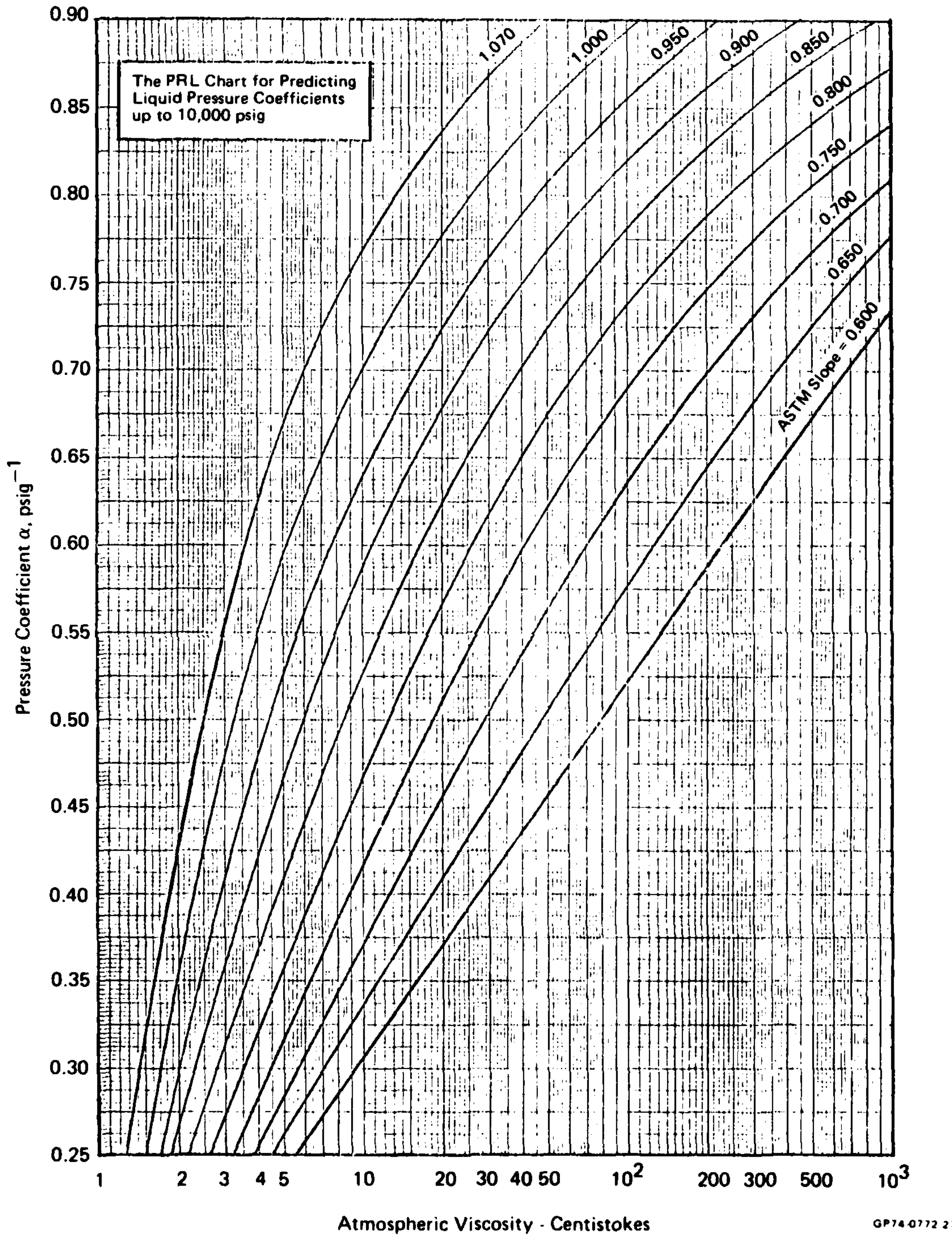


FIGURE B-2
FLUID FACTOR CHART

Professor Klause developed coefficients for various ASTM slope values to be used in his pressure coefficient equation.

$$\alpha = A + B \log V_o + C (\log V_o)^2 \tag{7}$$

where A, B and C are constants at the various slopes as shown below.

ASTM Slope	A	B	C
0.600	0.0878	0.2187	-0.0009
0.650	0.0578	0.2953	-0.0176
0.700	0.0425	0.3760	-0.0395
0.750	0.0379	0.4519	-0.0626
0.800	0.0546	0.5045	-0.0809
0.850	0.0720	0.5630	-0.1046
0.900	0.0947	0.6319	-0.1368
0.950	0.1064	0.7290	-0.1863
1.000	0.1384	0.8042	-0.2364
1.070	0.1423	1.0490	-0.4186

The writer then undertook the task to develop computer equations which calculate first the ASTM slope between two adjacent input data points, then the pressure correction coefficient. It may be recognized that when the user inputs actual fluid viscosity-temperature data points, these may not all lie on the theoretical straight line of the ASTM chart. Therefore the approach for finding the ASTM slope for the fluid at any given temperature is to take the input viscosity-temperature point on either side of the given temperature and in this region use the slope between the two input data points. If extrapolation is required for a given temperature, the 2 input points on either side of the given temperature are used to find the slope.

Professor Klause's work was based on the pre 1974 ASTM charts; therefore the equations for pressure correction use the constant of 0.6.

The writer selected the viscosity range of 5 centistokes to 1000 centistokes to calculate the reference slope because this is the range for which most of the anticipated work would occur. However any other range could just as well have been used. Using the MacCoull equation with a 0.6 constant.

$$\log \log (\nu + .6) = A - B \log T \tag{8}$$

log - logarithm to base 10

ν - kinematic viscosity in centistokes

A and B -Constants

T - Temperature in degrees Rankine

Using the 2 input data points on either side of the given temperature the equation is solved for A and B constants. Since viscosity and temperature are known, this is just an algebraic solution of equation (8).

With A and B known, the equation may again be solved, this time for T.

$$\log T = \frac{\log \log (\nu + .6) + A}{B} \tag{9}$$

OR

$$T = 10^{\left(\frac{(\log \log (\nu + .6) + A)}{B}\right)} \tag{10}$$

With the reference values for viscosity of 5 and 1000 centistokes, this equation becomes

$$T_5 = 10^{\left(\frac{.125989 + A}{B}\right)} \tag{11}$$

$$T_{1000} = 10^{\left(\frac{-.477159 + A}{B}\right)} \tag{12}$$

Using an ASTM chart, the measured ΔY value for the viscosity difference between 5 and 1000 centistokes is 6.65 inches.

An equation was worked out for distance along the temperature (X) axis to be

$$\Delta X = 65.10979 \times \left[\log \frac{T_5}{100} + 1 - \log \frac{T_{1000}}{100} + 1\right] \qquad (13)$$

The slope may now be calculated

as slope $= S = \frac{Y}{X} = \frac{\Delta Y}{\Delta X} = \frac{6.65}{\Delta X}$

A curve fit was made of Professor Klause's A, B and C constants versus ASTM slope as defined in equation (7).

These are as follows:

$A = 3.23523 - 11.3886 x S + 13.1735 X S^2 - 4.8881 x S^3$

$B = 5.33425 + 19.9521 x S - 23.9448 x S^2 + 10.155 x S^3$

$C = 3.35452 - 13.1273 x S + 17.1712 x S^2 - 7.6551 x S^3$

These values are substituted into equation (7) along with the atmospheric viscosity for calculation of the pressure coefficient. The equations developed in this Appendix B are the basis for VISD, INTERP and LAGRAN subroutines.

REFERENCES

B1. Wright, W. A., "An Improved Viscosity - Temperature Chart for Hydrocarbons," Journal of Materials, Vol 4, No. 1, 1969, pp 19-27

B2. Manning, R.E., "Computational Aids for Kinematic Viscosity Conversions from 100 and 210°F to 40 and 100°C," Journal of Testing and Evaluation, JTEVA, Vol 2, No. 6, November 1974

B3. Annual Book of ASTM Standards, American Society for Testing Materials, Philadelphia, 1974; Part 23

APPENDIX C

3 March 1975

Revised 30 November 1979

ENERGY LOSS IN TEES AND CROSSES

AND

RESISTANCE IN BENDS

BY

R. E. YOUNG

BASED ON ARTICLE

PRESSURE LOSS IN ELBOWS

AND DUCT BRANCHES

BY

ANDREW VAZSONYI

APRIL 1944 ASME TRANSACTIONS

Energy Loss in Tees and Crosses

In addition to the normal friction losses in a tee or cross there is an additional energy loss due to uniting or separating the fluid.

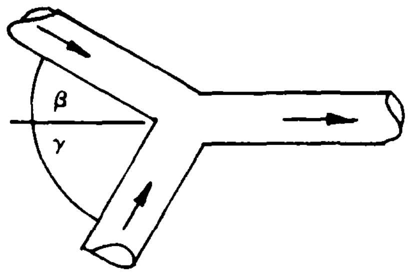


Figure C-1
Uniting Flow

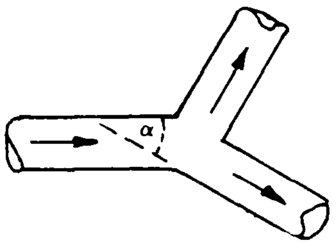


Figure C-2
Separating Flow

The total pressure loss will depend on at least the following quantities:

(1)	Pipe diameters	(3 variables)
(2)	The fluid velocities	(3 variables)
(3)	The angles of the branches	(2 variables)
(4)	Roughness of walls	(1 variable)
(5)	Viscosity of the fluid	(1 variable)

Using the continuity equation and similarity consideration, four of these variables can be eliminated and the remaining six can be listed as follows:

(1)	Ratios of exit areas to inlet area	(2 variables)
(2)	Ratio of velocities in 2 of the pipes	(2 variables)

(3) The branching angles (1 variable)

(4) Friction factor (1 variable)

The pressure loss due to friction can be determined using the friction factor. The pressure loss due to mixing is independent of the wall roughness. The formulas were derived for the mixing loss and then coefficients were added to the formulas which made them fit test data.

The following equations were developed to predict the mixing loss.

SEPARATING FLOW

$$Z\, h_{01} = \lambda_1 V_0^{\,2} + (2\lambda_1 - \lambda_2)\, V_1^{\,2} - 2\lambda_2\, V_0 V_1 \cos\alpha' \qquad \text{(I)}$$

UNITING FLOW

$$Z\, h_{10} = \lambda_3\, V_1^{\,2} + V_0^{\,2} - 2\, V_0 \left(V_1 \frac{Q_1}{Q_0} \cos\beta' + V_2 \frac{Q_2}{Q_0} \cos\gamma' \right) \qquad \text{(II)}$$

<u>Theory of Development of Equations</u>

The loss in the fitting will obviously depend on the wall roughness besides the geometry of the fitting. By dividing the losses into two groups "friction loss" and "mixing loss", a general equation can be written expressing these quantities for flow from point 1 to point 2.

$$\overbrace{P_1 + \frac{1}{2}\rho\, V_1^{\,2}}^{(1)} = \overbrace{P_2 + \frac{1}{2}\rho\, V_2^{\,2}}^{(2)} + \overbrace{\frac{F_1 L_1 P_1}{A_1} \cdot \frac{1}{2} P\, V_1^{\,2}}^{(3)} + \overbrace{\frac{F_2 L_2 P_2}{A_2} \cdot \frac{1}{2}\rho\, V_2^{\,2}}^{(4)} + \overbrace{h\rho}^{(5)} \qquad \text{(III)}$$

(1) and (2) are expressions of total pressure (static + dynamic) at points 1 and 2 in the system. (3) and (4) are expressions for the friction loss and (5) is the expression for the mixing loss. It should be noted that point 1 is normally the base or starting point, therefore L_1 in expression (3) is zero making expression (3) zero. The friction loss is then expressed by (4) using the actual center line length. Equation (III) can now be written as:

$$\underbrace{P_1 - P_2}_{\substack{\text{Change in}\\ \text{Static}\\ \text{Pressure}}} - \underbrace{\frac{1}{2}\rho\,(V_2^{\,2} - V_1^{\,2})}_{\substack{\text{Change in}\\ \text{Kinetic}\\ \text{Pressure}}} = \underbrace{\frac{1}{2}\rho \left(\frac{F_2 L_2 P_2}{A_2} V_2^{\,2} \right)}_{\text{Friction Loss}} + \underbrace{h\rho}_{\substack{\text{Mixing}\\ \text{Loss}}} \qquad \text{(IIIa)}$$

Experiments have shown that mixing loss is relatively unaffected by a change in wall roughness.

The mixing loss is then written without the friction loss term.

$$2\ h = 2\ (\frac{P_1 - P_2}{\rho}) - (V_2^2 - V_1^2)$$

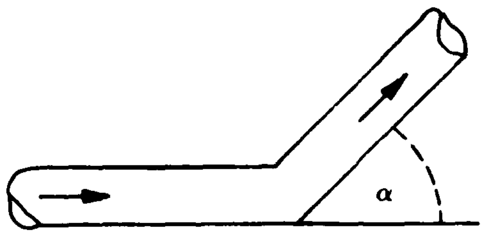


FIGURE C-3

Writing the momentum equation for Figure C-3 gives

$$P_1 A_1 + \rho\ V_1^2 \frac{A_1}{A_2} \cos \alpha = P_2 A_2 + \rho\ V_2^2 \qquad \frac{A_1}{A_2} = \frac{V_2}{V_1}$$

$$\frac{P_1 - P_2}{\rho} = V_2^2 = V_2^2 = V_1 V_2 \cos \alpha$$

$$2\ h = 2\ V_2^2 - 2V_1 V_2 \cos\ - V_2^2 + V_1^2$$

$$2\ h = V_1^2 + V_2^2 - 2V_1 V_2 \cos \alpha$$

Since assumptions have been made in deriving this equation, it is probably not exact and should be written as:

$$2\ h = \xi\ (V_1^2 + V_2^2 - 2V_1\ V_2 \cos \alpha)$$

where ξ is an experimental coefficient. The elbow acts as a diffuser and little turbulence occurs in the passage. Diffuser losses have been successfully correlated using the expression

$$2\ h = (1 - n)\ (V_2^2 - V_1^{\ 2})$$

where n is the efficiency of the diffuser.

Combining the two different expressions for elbow loss.

$$2\ h = \xi\ (V_1^{\ 2} + V_2^{\ 2} - 2V_1V_2 \cos\alpha) + (1-n)\ (V_2^{\ 2} - V_1^{\ 2})$$

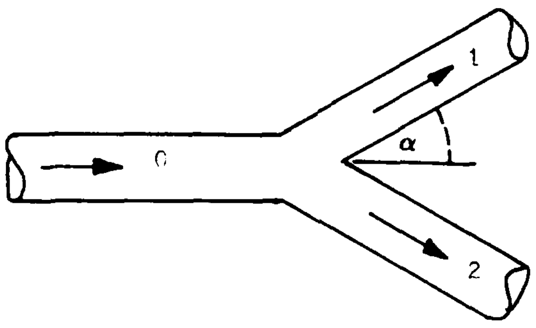
or

$$2\ h = \lambda_4V_1^{\ 2} - 2\lambda V_1V_2 \cos\alpha + (2\lambda-\lambda_4)\ V_2^{\ 2}$$

When $V_2 = 0$, all the entering kinetic energy will be dissipated so $\lambda_4 = 1$.

The branching pipe can be considered as 2 elbows:

<u>FOR SEPARATING FLOW</u>



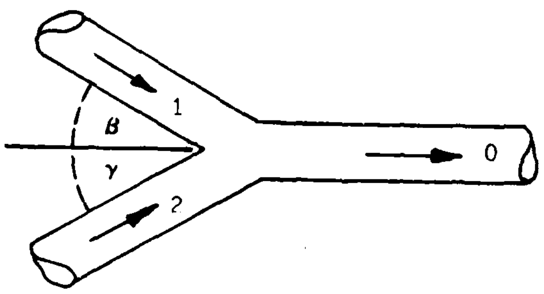
The elbow formula is applied to give:

$$2\ h_{01} = \lambda_1V_0^{\ 2} + (2\lambda_1-\lambda_2)\ V_1^{\ 2} - 2\lambda_0V_0V_1 \cos\alpha'$$

α' is the average angle

λ_1 and λ_2 are determined experimentally

FOR UNITING FLOW



The momentum equation is applied to the leaving section with the inlet pipes acting as diffusers giving:

$$2\ h_{10} = \lambda_3 V_1^{\ 2} + V_0^{\ 2} - 2V_0\ (\frac{V_1 Q_1}{Q_0} \cos \beta' + \frac{V_2 Q_2}{Q_0} \cos \gamma')$$

The equations developed above can be used for any angle of uniting or separating flow with the aid of Figure C-4 for separating flow and Figure C-5 for uniting flow.

Probably the most common section used for separating flow and uniting flow in fluid systems is the standard tee. For this type of fitting the above equations can be simplified for a given application.

SEPARATING FLOW FOR STANDARD TEE

For $\alpha = 90°$, $\lambda_1 = 1.0$ and $\lambda_2 = 0.9$

$\frac{\alpha'}{\alpha} = .85 \qquad \alpha' = .85\alpha = (.85)\ (90") = 76.5°$

$\cos \alpha' = \cos 76.5° = .233$

Equation I can be written as:

$$2\ h_{01} = (1)V_0^{\ 2} + (2(1) - .9)\ V_1^{\ 2} - 2\ (.9)\ V_0 V_1\ (.233)$$

$$2\ h_{01} = V_0^{\ 2} + 1.1\ V_1^{\ 2} - .42\ V_0 V_1$$

$$h_{01} = .5\ V_0^{\ 2} + .55\ V_1^{\ 2} - .21\ V_0 V_1 \qquad \text{(Ia)}$$

UNITING FLOW FOR STANDARD TEE

For β and γ = 90° $\qquad\qquad \lambda_3 = .6$

$$\frac{\beta'}{\beta} = \frac{\gamma'}{\gamma} = .85 \qquad \gamma' = \beta' = .85\gamma = .85\beta = .85\,(90°) = 76.5°$$

$$\cos\gamma = \cos\beta' = \cos 76.5° = .233$$

$$2\,h_{10} = .6\,V_1^2 + V_0^2 - 2\,V_0\left(V_1\frac{Q_1}{Q_0}(.233) + \frac{V_2Q_2}{Q_0}(.233)\right)$$

$$h_{10} = .3\,V_1^2 + .5\,V_0^2 - .233\,V_0\left(\frac{V_1Q_1 + V_2Q_2}{Q_0}\right) \qquad \text{(IIa)}$$

Note: Equations (Ia) and (IIa) are the mixing losses in the standard tees and do not include friction losses. These equations are also used to determine h_{02} and h_{20} by changing the subscripts.

APPLICATION OF EQUATIONS

The mixing losses in Equations (Ia) and (IIa) are now functions of only velocity and flow rate.

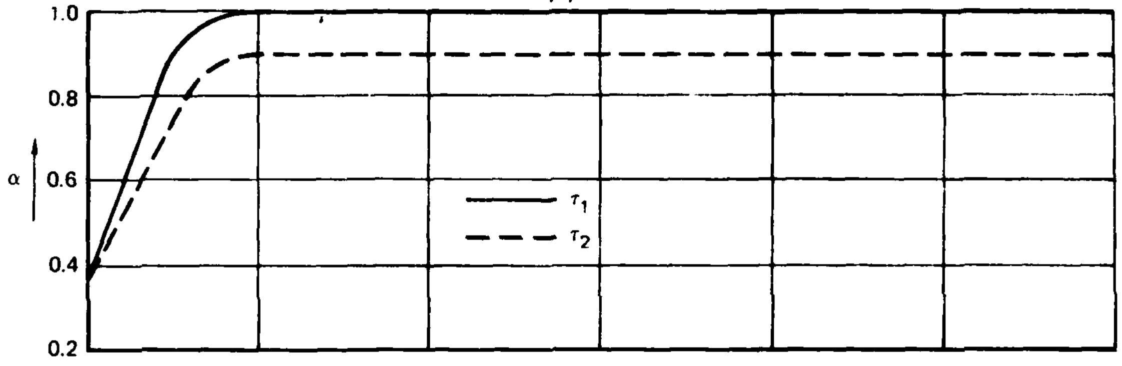
In these equations, the units are as follows:

	Quantity	Units
Head loss	h	FT^2/SEC^2
Velocity	V	FT/SEC
Volume flow rate	Q	FT^3/SEC
Weight density	W	LB/FT^3
Mass density	ρ	$LB\text{-}SEC^2$ —— $= \frac{w}{g}$
Gravitational constant	g	

To convert head loss, h, to PSI, multiply h by $\frac{\text{mass density}}{144}$ or can be written as:

$$PSI = \frac{h\,\rho}{144}$$

Pressure Loss in Branch with Separating Flow
(a)



Effective Angle of Deflection as a Function
of Angle of Deflection
(b)

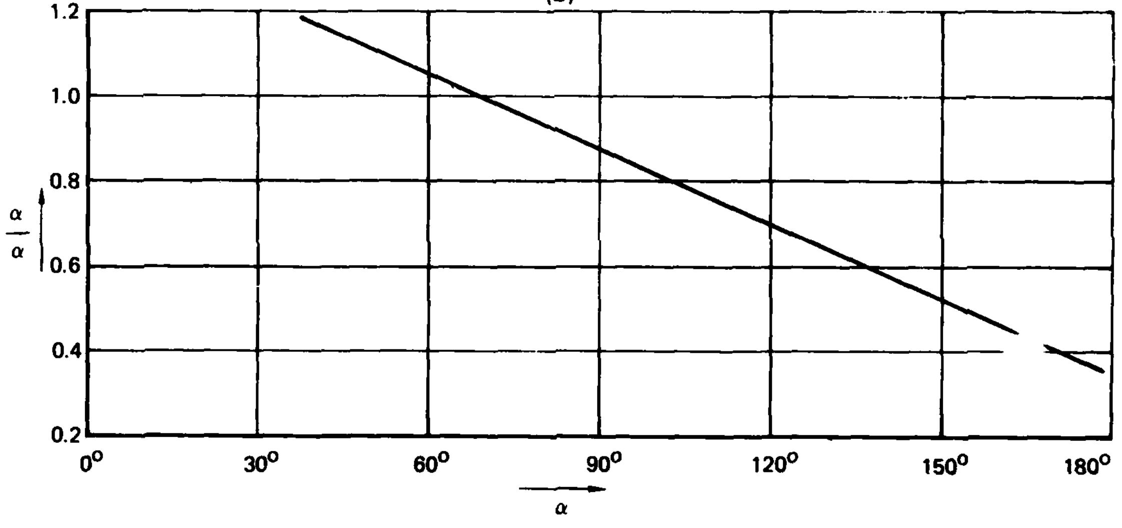


FIGURE C-4
USED WITH EQUATION FOR SPEARATING FLOW

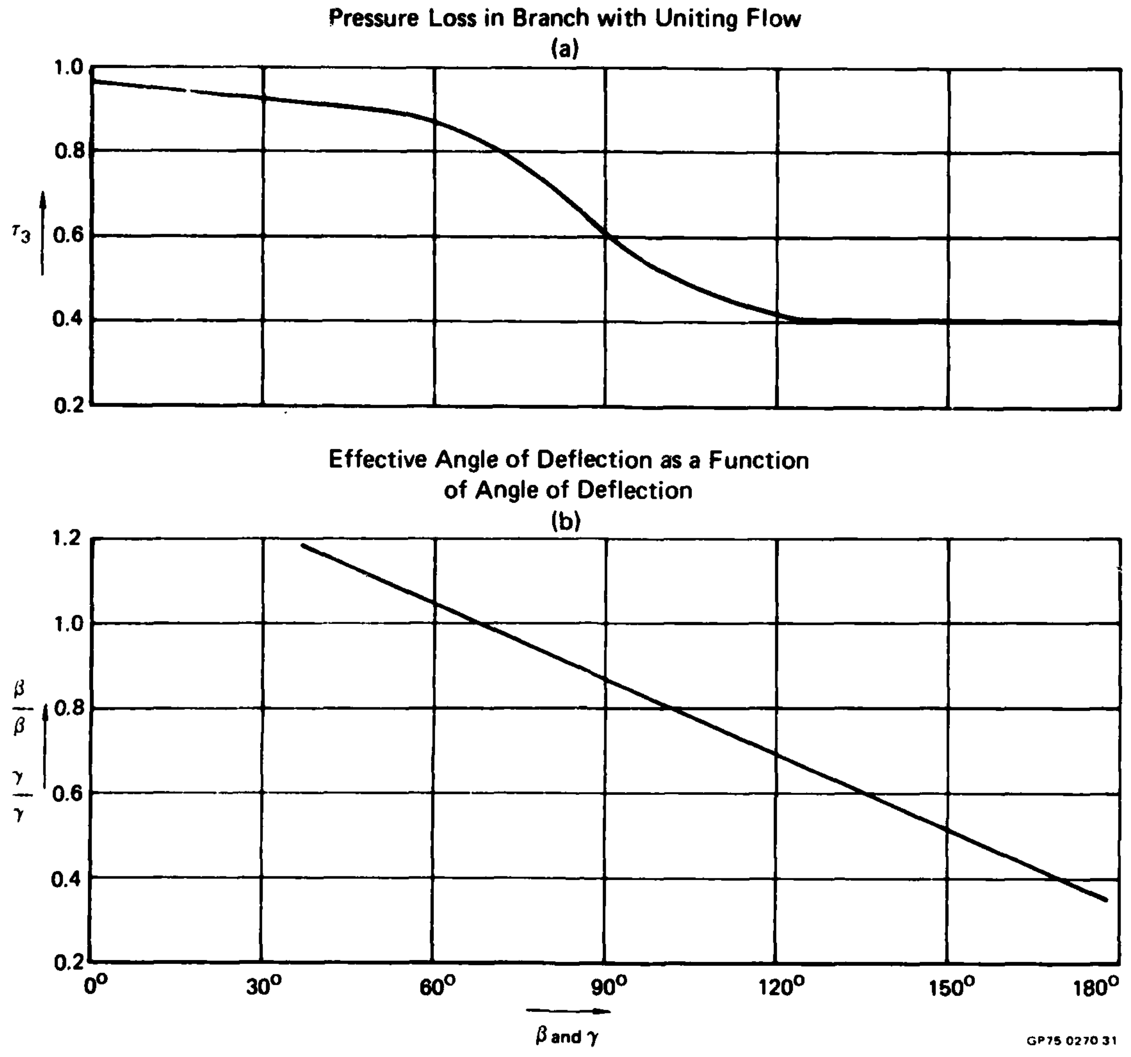


FIGURE C-5
USED WITH EQUATION FOR UNITING FLOW

Resistance of Bends

The total resistance of any bend is comprised of two parts. The first of these parts is a frictional resistance due to the length of the bend and the second part is an energy loss due to the change in flow direction. SSFAN uses the center line length of the bent tube to calculate the frictional resistance of the tube. This is accomplished in subroutine SPORT.

The energy loss portion of a tube's resistance is due to three factors: bend angle, severity of the bend, and cross sectional configuration of the bend. The method used to obtain the energy loss portion of a tube or hose resistance in the SSFAN Program is obtained from Reference C1. Reference C1 interrelates the effect of each energy loss factor to the total energy loss coefficient of the bend through the simple formula:

$$f = A_1 + B_1 + C_1$$

Where: f = energy loss coefficient of the bend

A_1 = effect of the bend angle

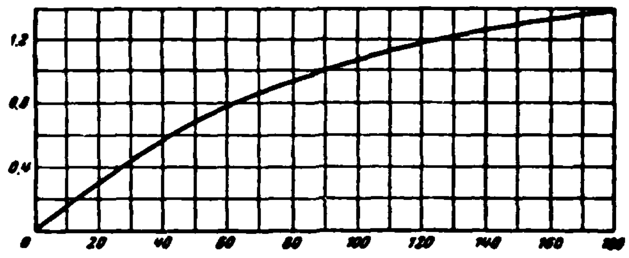
B_1 = effect of the bend severity

C_1 = effect of the bend cross section

SSFAN models energy loss coefficients in tubes and hoses with smooth bends of constant circular cross section through use of the above formula. The coefficient of each bend in the tube or hose is calculated then summed with the coefficients of the other bends to give the total energy loss coefficient of the element.

Figure C-1 illustrates the relationship between the variable A_1 and the angle of bend of the tube. SSFAN uses a computer generated fourth degree polynomial curve fit to calculate values of A_1 for bends of 180° or less and approximates the contribution of bends greater than 180° by the linear equation:

$$A1 = 1.39 + (DEG - 180) \times 3.333 \times 10^{-3}$$



BEND ANGLE (DEG)

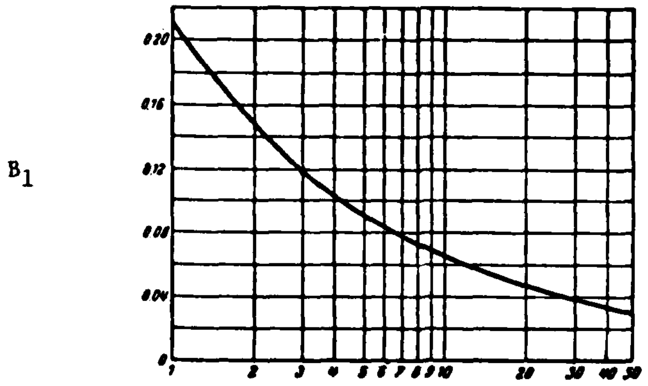
FIGURE C-1

Similarly, calculation of the variable B_1 is achieved through different equations depending upon which region of bend ratio (bend radius/inside dia) the tube bends fall in. A curve fit of Figure C-2 for bend ratios between 1 and 30 yields the power function.

$$B_1 = .206982 \times \text{Bend Ratio}^{-.49421}$$

For bend ratios between 30 and 50 the relationship is given by the equation

$$B_1 = .0385411 - (\text{Bend Ratio} - 30) \times .0004$$



BEND RATIO

FIGURE C-2

If no bend radius for the tube or hose is input on the element data card, the default value of the bend ratio is 3.56 if a tube is being calculated and 7.21 if a hose is under consideration.

Since all bends are assumed to be of constant circular cross section, the variable C_1 requires no calculation because the configuration does not vary. Reference A shows that for circular cross section conduits

$$C_1 = 1.00$$

REFERENCES

C1. USAEC-TR-6630, "Handbook of Hydraulic Resistance", translated from Russian and printed in Jerusalem by S. Monson, 1966

*U.S.Government Printing Office: 1980 — 657-084/50

ATE

LMED

-8